



## ΕΘΝΙΚΟΝ & ΚΑΠΟΔΙΣΤΡΙΑΚΟΝ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**Τίτλος Έργου: «StochSocS: Συστήματα σε Ψηφίδα για Παράλληλη  
Στοχαστική Προσομοίωση Βιολογικών Δικτύων στη Βιολογία  
Συστημάτων»**

**(Αριστεία II κωδικός 3828 και Κ.Α. 70/3/12367)**

**ΠΑΡΑΔΟΤΕΟ 1.2**

**Σχεδίαση Αρχιτεκτονικής SoC για τον NRM-SSA**

ΑΘΗΝΑ

ΜΑΪΟΣ 2015



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
Πρόγραμμα για την ανάπτυξη

ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Δηλώνω υπεύθυνα και εν γνώσει των συνεπειών του νόμου ότι το Παραδοτέο «Σχεδίαση Αρχιτεκτονικής SoC για τον NRM-SSA» αποτελεί πρωτότυπο έργο και δεν έχουν χρησιμοποιηθεί άλλες πηγές πέρα από αυτές στις οποίες γίνεται αναφορά. Όπου υπάρχει αναπαραγωγή δεδομένων, πινάκων και παραθεμάτων, αυτό δηλώνεται ρητά.

Ηλίας Μανωλάκος



## Περιεχόμενα

1	ΕΙΣΑΓΩΓΗ.....	6
1.1	Η Εξέλιξη της Βιολογίας.....	6
1.2	Βιολογικά Δίκτυα – Μέθοδοι Προσομοίωσης.....	6
1.3	Στόχος - Σκοπός.....	7
1.4	Διάρθρωση Παραδοτέου.....	8
2	ΜΕΘΟΔΟΙ ΣΤΟΧΑΣΤΙΚΗΣ ΠΡΟΣΟΜΟΙΩΣΗΣ.....	10
2.1	Μοντελοποίηση Βιολογικών Δικτύων.....	10
2.2	Ο Αλγόριθμος NRM των Gibson και Bruck.....	11
2.3	Η Διαφοροποίηση του NRM-SSA έναντι του FRM-SSA.....	12
2.4	Ο Γράφος Εξαρτήσεων.....	12
3	ΠΡΟΔΙΑΓΡΑΦΕΣ ΣΥΣΤΗΜΑΤΟΣ.....	15
3.1	Προδιαγραφές της παραμετρικής αρχιτεκτονικής NRM SoC.....	15
3.2	Λειτουργία Συστήματος και Ροή Εργασιών.....	16
4	ΑΡΧΙΤΕΚΤΟΝΙΚΗ NRM ΣΥΣΤΗΜΑΤΟΣ ΣΕ ΨΗΦΙΔΑ.....	19
4.1	Αρχιτεκτονική Πυρήνα NRM-SSA.....	19
4.2	Λειτουργία Πυρήνα και Ροή Δεδομένων.....	20
4.3	Διαφοροποίηση Πυρήνων FRM - NRM.....	22
5	ΑΡΧΙΤΕΚΤΟΝΙΚΗ NRM ΕΠΕΞΕΡΓΑΣΤΙΚΗΣ ΜΟΝΑΔΑΣ.....	23
5.1	NPM Processing Unit - NPU.....	23
5.2	NRM Processing Control Unit – NPCU.....	25
5.3	NRM Processing Element – NPE.....	27
5.4	Dependency Graph (DG_MS).....	28
5.5	Ροή Δεδομένων – NPU Datapath.....	31
6	Η ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ - NRM PROCESSING ELEMENT.....	34
6.1	Βασική Λειτουργία και Ροή Δεδομένων.....	34
6.2	Πίνακας Χρόνων Ενεργοποίησης Αντιδράσεων.....	35
6.3	Δέντρο Συγκριτών – Comparator Tree (CoT).....	37
6.4	Μονάδα Επανυπολογισμού – Recalculation Unit (RU).....	38
7	ΕΠΙΔΟΣΕΙΣ - ΣΤΑΤΙΣΤΙΚΑ.....	43
7.1	Θεωρητική Ανάλυση Επιδόσεων.....	43
7.2	Στατιστικά Υλοποίησης.....	45
8	ΕΠΙΚΥΡΩΣΗ ΣΧΕΔΙΑΣΗΣ.....	51
9	ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	53

ΒΙΒΛΙΟΓΡΑΦΙΑ.....	55
-------------------	----



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



**ΕΣΠΑ**  
**2007-2013**  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

## Κατάλογος Εικόνων

Εικόνα 1: Ενδεικτικό παράδειγμα γράφου εξαρτήσεων αντιδράσεων [27].....	13
Εικόνα 2: Το ιστόγραμμα των επηρεαζόμενων αντιδράσεων του ASYN βιομοντέλου [28]. Ο άξονας y απεικονίζει τον αριθμό των αντιδράσεων του μοντέλου που έχουν το ίδιο πλήθος επηρεαζόμενων αντιδράσεων (D).....	13
Εικόνα 3: Ο χρήστης δίνοντας ως είσοδο το SBML [35] αρχείο του βιομοντέλου και το αρχείο παραμέτρων, θα λάβει ως έξοδο τα αποτελέσματα της επιθυμητής προσομοίωσης.....	17
Εικόνα 4: Η γενική αρχιτεκτονική δομή του NRM SoC. Το πλήθος των πηρύνων είναι παραμετρος του συστήματος.....	19
Εικόνα 5: Η αρχιτεκτονική ενός πυρήνα NRM-SSA.....	21
Εικόνα 6: Το RTL σχηματικό της μονάδας NPU.....	23
Εικόνα 7: Η αρχιτεκτονική της μονάδας NRM Processing Unit (NPU).....	24
Εικόνα 8: Το RTL σχηματικό της μονάδας NPCU.....	25
Εικόνα 9: Το RTL σχηματικό της μονάδας NPE.....	27
Εικόνα 10: Το RTL σχηματικό της μονάδας DG_MS.....	29
Εικόνα 11: Ενδεικτικό παράδειγμα λειτουργίας γράφου εξαρτήσεων.....	30
Εικόνα 12: Η αρχιτεκτονική της μονάδας NRM Process Element (NPE).....	35
Εικόνα 13: Ενδεικτικό παράδειγμα του πίνακα T και του δέντρου συγκριτών (CoT) ύψους H=3.....	36
Εικόνα 14: Το RTL σχηματικό της μονάδας Recalculation Unit (RU).....	39
Εικόνα 15: Η αρχιτεκτονική της μονάδας Recalculation Unit.....	41
Εικόνα 16: Η ανάλυση της καθυστέρησης λόγω διοχέτευσης του πυρήνα NRM.....	43
Εικόνα 17: Θεωρητικά αναμενόμενη διεκπεραιωτική ικανότητα (Reaction Throughput) του NRM SoC συναρτήσει του πλήθους των αντιδράσεων (m) και του μέσου όρου επηρεαζόμενων αντιδράσεων (Daver). Συχνότητα λειτουργίας 200MHz.....	45
Εικόνα 18: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM Core για m = 512/1024 και διαφορετικά Daver.....	47
Εικόνα 19: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM Core για m = 2048/4096 και διαφορετικά Daver.....	47
Εικόνα 20: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM SoC για m = 512/1024 και διαφορετικά Daver.....	49
Εικόνα 21: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM SoC για m = 2048/4096 και διαφορετικά Daver.....	49
Εικόνα 22: Σύγκριση των προσομοιωτών NRM NoC και COPASI.....	52

## Κατάλογος Πινάκων

Πίνακας 1: Όλα τα σήματα εισόδου και εξόδου του NRM Core, καθώς και οι γενικές παράμετροι του πυρήνα.	20
Πίνακας 2: Τα σήματα εισόδου και εξόδου της μονάδας NPU.....	23
Πίνακας 3: Τα σήματα εισόδου και εξόδου της μονάδας NPCU.....	26
Πίνακας 4: Τα σήματα εισόδου και εξόδου της μονάδας NPE.....	27
Πίνακας 5: Τα σήματα εισόδου και εξόδου της μονάδας DG_MS.....	28
Πίνακας 6: Το πλήθος των απαιτούμενων κύκλων ρολογιού ( <i>LCoT</i> ) συναρτήσει του πλήθους των αντιδράσεων <i>m</i> και του ύψους του δέντρου <i>H</i> .....	38
Πίνακας 7: Τα σήματα εισόδου και εξόδου της μονάδας Recalculation Unit (RU).....	40
Πίνακας 8: Οι απαιτούμενοι κύκλοι ρολογιού των διακλαδώσεων της RU.....	42
Πίνακας 9: Χαρακτηριστικά του FPGA XC7K325T.....	45
Πίνακας 10: Τα στατιστικά υλοποίησης του NRM SoC ( <i>C</i> = 1) για <i>m</i> = <i>n</i> = 512 και διαφορετικά <i>Daver</i> .....	46
Πίνακας 11: Τα στατιστικά υλοποίησης του NRM SoC ( <i>C</i> = 1) για <i>m</i> = <i>n</i> = 1024 και διαφορετικά <i>Daver</i> .....	46
Πίνακας 12: Τα στατιστικά υλοποίησης του NRM SoC ( <i>C</i> = 1) για <i>m</i> = <i>n</i> = 2048 και διαφορετικά <i>Daver</i> .....	46
Πίνακας 13: Τα στατιστικά υλοποίησης του NRM SoC ( <i>C</i> = 1) για <i>m</i> = <i>n</i> = 4096 και διαφορετικά <i>Daver</i> .....	46
Πίνακας 14: Τα στατιστικά υλοποίησης του NRM SoC ( <i>C</i> = 8) για <i>m</i> = <i>n</i> = 512 και διαφορετικά <i>Daver</i> .....	48
Πίνακας 15: Τα στατιστικά υλοποίησης του NRM SoC ( <i>C</i> = 8) για <i>m</i> = <i>n</i> = 1024 και διαφορετικά <i>Daver</i> .....	48
Πίνακας 16: Τα στατιστικά υλοποίησης του NRM SoC ( <i>C</i> = 4) για <i>m</i> = <i>n</i> = 2048 και διαφορετικά <i>Daver</i> .....	48
Πίνακας 17: Τα στατιστικά υλοποίησης του NRM SoC ( <i>C</i> = 2) για <i>m</i> = <i>n</i> = 4096 και διαφορετικά <i>Daver</i> .....	49



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
Επένδυση στην κοινωνία της γνώσης  
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
Πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# 1 ΕΙΣΑΓΩΓΗ

## 1.1 Η Εξέλιξη της Βιολογίας

Μια από τις σημαντικότερες και ραγδαία εξελισσόμενες επιστήμες της εποχής μας, είναι η βιολογία. Η βιολογία εξετάζει τους μηχανισμούς της ζωής και πώς αυτοί αλληλεπιδρούν. Η βαθύτερη κατανόηση των μηχανισμών αυτών θα οδηγήσει στη αποτελεσματικότερη αντιμετώπιση των ασθενειών που μαστίζουν το πλανήτη αλλά και τη βέλτιστη αξιοποίηση των φυσικών του πόρων [1]. Καθώς η βιοτεχνολογία αναπτύσσεται, ο άνθρωπος έχει τη δυνατότητα να μελετήσει τα βιολογικά συστήματα σε μεγαλύτερη κλίμακα αλλά και με μεγαλύτερη διακριτική ικανότητα. Η βιολογία έτσι επιμερίστηκε σε κλάδους, όπως η μικροβιολογία, η γενετική, και συνδέθηκε άρρικτα και με συναφείς επιστήμες, όπως η βιοχημεία, η βιοφυσική, αλλά και η βιοστατιστική, κ.α. [2]. Κάθε ένας από τους κλάδους αυτούς επικεντρώνεται σε μια πτυχή της επιστήμης, αλλά όλοι μαζί συνθέτουν τη κοινή ευρεία βάση που αποκαλούμε «σύγχρονη βιολογία».

Η σύγχρονη βιολογία στηρίζεται σε μεγάλο βαθμό στη Πληροφορική. Είναι γεγονός ότι οι σύγχρονες βιολογικές μελέτες μεγάλης κλίμακας παράγουν τεράστιο φόρτο δεδομένων ("big data") που απαιτούν ισχυρές υπολογιστικές μονάδες για την επεξεργασία τους για την εξαγωγή χρήσιμων συμπερασμάτων [3]. Δύο νέοι συναφείς κλάδοι που αναπτύχθηκαν και θα ανθίσουν την τρέχουσα δεκαετία είναι η *Υπολογιστική Βιολογία* [4] και η *Βιοπληροφορική* [5]. Η Υπολογιστική Βιολογία επικεντρώνεται στην ανάπτυξη θεωρητικών μεθόδων και μαθηματικών μοντέλων για τη μελέτη της οργάνωσης, λειτουργίας και αλληλεπίδρασης βιολογικών συστημάτων σε διαφορετικά επίπεδα αφαίρεσης. Ενώ, η Βιοπληροφορική ασχολείται κυρίως με την έρευνα και την ανάπτυξη κατάλληλων εργαλείων και προσεγγίσεων για την αποδοτικότερη μελέτη και συνδιασμένη ανάλυση και ερμηνεία των παραγόμενων δεδομένων βιολογικών πειραμάτων. Για τους σκοπούς αυτούς, η Υπολογιστική Βιολογία και η Βιοπληροφορική αξιοποιούν μεθόδους κλάδων όπως η ανάλυση αλγορίθμων, οι βάσεις δεδομένων, η εξόρυξη γνώσης [6], η υπολογιστική θεωρία, η θεωρία ελέγχου [7], η θεωρία συστημάτων, η μικροηλεκτρονική [8], η μοντελοποίηση και προσομοίωση συστημάτων, η τεχνητή νοημοσύνη [9], η αναγνώριση προτύπων [10], η επεξεργασία σημάτων, η ανάπτυξη συστημάτων υλικού κ.α.

## 1.2 Βιολογικά Δίκτυα – Μέθοδοι Προσομοίωσης

Η σύγχρονη βιολογία απαιτεί την ανάπτυξη υπολογιστικών μοντέλων που να δίνουν τη δυνατότητα μελέτης in-silico (με χρήση υπολογιστή) της δομής και λειτουργίας τόσο των κυττάρων όσο και των πολυκυττάρων οργανισμών [11] με σκοπό την εκτέλεση πειραμάτων μέσω υπολογιστή. Ενισχύεται λοιπόν συνεχώς η ανάγκη για ανάπτυξη κατάλληλων μεθόδων και εργαλείων που θα μπορούν να προσομοιώνουν τη δυναμική συμπεριφορά των δικτύων αλληλεπιδράσεων βιομορίων σε διάφορα επίπεδα αφαίρεσης, κάτι που αποτελεί το καθ'εξοχήν αντικείμενο μελέτης του πλέον πρόσφατου κλάδου της *Συστημικής Βιολογίας* (Systems Biology) [12] [13].

Μέχρι σήμερα γνωρίζουμε δύο βασικές μεθόδους προσομοίωσης της δυναμικής συμπεριφοράς δικτύων βιοχημικών αντιδράσεων. Η πρώτη βασίζεται στη χρήση Συνήθων Διαφορικών Εξισώσεων (ODEs), ενώ η δεύτερη στηρίζεται σε αλγόριθμους στοχαστικής προσομοίωσης (Stochastic Simulation). Η χρήση Συνήθων Διαφορικών Εξισώσεων (ODEs) στηρίζεται σε δύο απλουστευτικές παραδοχές που δε συναντάμε στη φύση [14]. Η πρώτη είναι ότι το βιολογικό σύστημα που μελετάμε έχει ντετερμινιστική συμπεριφορά. Οι ντετερμινιστικές μέθοδοι παράγουν πάντα την ίδια έξοδο, αγνοώντας στοχαστικά φαινόμενα του περιβάλλοντος όπως ο "βιολογικός θόρυβος". Η δεύτερη παραδοχή αφορά την ποσοτικοποίηση των μοριακών ειδών που

αλληλεπιδρούν. Ενώ γνωρίζουμε ότι στη φύση το πλήθος ενός μοριακού είδους είναι πάντα ένας φυσικός αριθμός και συνεπώς κινείται σε ένα φάσμα διακριτών τιμών, στις προσομοιώσεις με διαφορικές εξισώσεις η συνθήκη αυτή δεν εξασφαλίζεται πάντα. Λαμβάνοντας υπόψιν τις δύο αυτές απλουστευτικές παραδοχές, συμπεραίνουμε ότι τα αποτελέσματα των ντετερμινιστικών προσομοιώσεων με τη χρήση ODEs ενδέχεται να διαφέρουν από τη πραγματική συμπεριφορά ενός βιολογικού μοντέλου, ιδιαίτερα όταν αυτό καλείται να περιγράψει έντονα στοχαστικά φαινόμενα, όπως π.χ. η μεταγραφή (transcription) γονιδίων σε ένα ρυθμιστικό δίκτυο (gene regulatory network).

Σε αντίθεση με τις ντετερμινιστικές μεθόδους, οι αλγόριθμοι στοχαστικής προσομοίωσης (stochastic simulation algorithms - SSA) δεν χρειάζεται να στηριχθούν σε απλουστευτικές υποθέσεις και άρα πληρούν καλύτερα τις προϋποθέσεις του φυσικού κόσμου [14]. Οι αλγόριθμοι αυτοί εμπεριέχουν τουλάχιστον ένα στοιχείο στοχαστικότητας (π.χ. χρησιμοποιούν γεννήτριες "τυχαίων" αριθμών) και χρησιμοποιούν διακριτές τιμές για τους μοριακούς πληθυσμούς που μπορεί να είναι και πολύ μικρές (low species counts) όπως στο κύτταρο. Αν και η στοχαστική προσομοίωση είναι πιο κοντά στην πραγματική συμπεριφορά ενός βιολογικού συστήματος, για να θεωρηθούν τα αποτελέσματα της ικανοποιητικά απαιτείται πολύ μεγάλος αριθμός στοχαστικών επαναλήψεων της προσομοίωσης ενός μοντέλου, με βάση μεθόδους Monte Carlo [15]. Ως εκ τούτου, η διαδικασία της στοχαστικής προσομοίωσης είναι πολύ χρονοβόρα και μπορεί να παράξει τεράστιο όγκο αποτελεσμάτων (εκατοντάδες με χιλιάδες καμπύλες μοριακών πληθυσμών). Τέλος, ένα ακόμα βασικό γνώρισμα των μεθόδων στοχαστικής προσομοίωσης είναι οι πεπερασμένες καταστάσεις στις οποίες το σύστημα μπορεί να μεταβεί. Η μετάβαση από τη μία κατάσταση στην άλλη πραγματοποιείται μέσω στοχαστικής διαδικασίας που μπορεί να μοντελοποιηθεί με Μαρκοβιανές αλυσίδες [16].

Στη παρούσα έρευνα σχεδιάζουμε αποδοτικά υπολογιστικά συστήματα υλικού για δύο δημοφιλείς αλγόριθμους στοχαστικής προσομοίωσης που βασίζονται στην Μαρκοβιανή προσέγγιση. Στο Παραδοτέο 1.1 μελετήσαμε τον αλγόριθμο του Gillespie, *First Reaction Method* (FRM) [14], ενώ στο τρέχον Παραδοτέο 1.2 μελετάμε τον πιο πρόσφατο αλγόριθμο των Gibson και Bruck, *Next Reaction Method* (NRM) [17]. Ενώ ο FRM-SSA μπορεί να παραλληλοποιηθεί αποδοτικά έχει μεγαλύτερη υπολογιστική πολυπλοκότητα  $O(m)$ , όπου  $m$  ο αριθμός των αντιδράσεων του βιολογικού δικτύου. Αντίθετα ο NRM-SSA έχει μικρότερη υπολογιστική πολυπλοκότητα,  $O(\log(m))$ , αλλά είναι πολύ πιο δύσκολο να παραλληλοποιηθεί γιατί στηρίζεται στη διατήρηση μιας κεντροποιημένης δομής δεδομένων (shared data structure) για να μειώσει τους απαιτούμενους υπολογισμούς. Είναι λοιπόν πολύ ενδιαφέρον αφού υλοποιηθούν και οι δύο αλγόριθμοι με υλικό ειδικού σκοπού να συγκριθούν οι υλοποιήσεις τους (Παραδοτέο 1.3).

### 1.3 Στόχος - Σκοπός

Μελετώντας τη σύγχρονη βιβλιογραφία και τις ήδη υπάρχουσες FRM – NRM αρχιτεκτονικές υλοποιήσεις, σκοπός μας είναι να αναπτύξουμε ένα αποδοτικό και πλήρως παραμετροποιημένο Σύστημα σε Ψηφίδα (System on Chip, SoC), χρησιμοποιώντας υλικό επαναπρογραμματιζόμενης λογικής (Field Programmable Gate Array, FPGA), το οποίο θα έχει τη δυνατότητα να προσομοιώνει παράλληλα και αποδοτικά πολύπλοκα δίκτυα βιομοριακών αλληλεπιδράσεων [18] [19] [20] [21] [22] [23] [24] [25]. Ο τελικός μας στόχος είναι να αναπτύξουμε ένα αποδοτικό υπολογιστικό σύστημα που θα ανταπεξέρχεται πλήρως στις απαιτήσεις της στοχαστικής προσομοίωσης μεγάλης κλίμακας αλλά επιπλέον θα δίνει στον χρήστη και τη δυνατότητα να καθορίζει όλες τις επιθυμητές παραμέτρους, τόσο σε επίπεδο προσομοίωσης (χρόνος προσομοίωσης, χρόνος δειγματοληψίας αποτελεσμάτων, πλήθος επαναλήψεων κ.α.) όσο και σε επίπεδο υπολογιστικού συστήματος



(πλήθος πυρήνων κτλ.).

Στο Παραδοτέο 1.1 έχει περιγραφεί η σχεδίαση και υλοποίηση συστήματος στοχαστικής προσομοίωσης βασιζόμενο στον αλγόριθμο First Reaction Method (FRM) του Gillespie. Η παράλληλη και πολυπύρηνη αρχιτεκτονική που ενσωματώνει, η δυνατότητα παραμετροποίησης των πυρήνων, καθώς και η άκρως ικανοποιητική απόδοσή τους, ανέδειξαν τα πλεονεκτήματα και τη σπουδαιότητα της χρήσης υλικού ειδικού σκοπού για εφαρμογές στοχαστικής προσομοίωσης βιολογικών δικτύων με τη χρήση FPGAs.

Το παρόν παραδοτέο επικεντρώνεται στην ανάπτυξη ενός ολοκληρωμένου υπολογιστικού συστήματος στοχαστικής προσομοίωσης βασιζόμενο στον αλγόριθμο Next Reaction Method (NRM) των Gibson και Bruck [17]. Σκοπός μας ήταν να επαναχρησιμοποιήσουμε μέρος του συστήματος που έχει ήδη σχεδιαστεί για τον FRM-SSA, ώστε να ελαχιστοποιηθεί ο χρόνος σχεδίασης, και να οδηγηθούμε σε μια αποδοτική λύση εφαρμόζοντας τεχνικές πλήρους σωλήνωσης (pipeline), παράλληλης επεξεργασίας (parallel processing) και απόκρυψης της καθυστέρησης (latency hiding) σε επίπεδο υλικού.

Εν τέλει, μετά την ολοκλήρωση και την επικύρωση των δύο συστημάτων, σκοπεύουμε να συγκρίνουμε τις δύο αρχιτεκτονικές υλοποιήσεις τόσο ως προς την αποδοτικότητα, επιδόσεις και απαιτήσεις υλικού, όσο και ως προς τον βαθμό στατιστικής ομοιότητας που παρουσιάζουν τα αποτελέσματά τους εκτελώντας τα ίδια πειράματα κάτω από τις ίδιες συνθήκες (Παραδοτέο 1.3).

## 1.4 Διάρθρωση Παραδοτέου

Στις ενότητες που ακολουθούν δίνεται έμφαση στη τεχνική περιγραφή του συστήματος στοχαστικής προσομοίωσης που σχεδιάσαμε και αναπτύξαμε, βασιζόμενο στον αλγόριθμο NRM-SSA των Gibson και Bruck. Αρχικά, γίνεται αναφορά στα βήματα του αλγορίθμου, στη συνέχεια ακολουθεί η πλήρης περιγραφή της δομής και της λειτουργικότητας του συστήματος σε ψηφίδα, παρουσιάζεται η θεωρητική ανάλυση των επιδόσεών του και τα αντίστοιχα στατιστικά υλοποίησης με βάση τους διαθέσιμους πόρους και, τέλος, παρουσιάζονται αποτελέσματα προσομοίωσης βιομοντέλων που επικυρώνουν τη λειτουργία του.

Συγκεκριμένα η διάρθρωση του παραδοτέου είναι η εξής:

*Ενότητα 2:* Παρουσιάζεται η μοντελοποίηση βιολογικών δικτύων, αναλύονται τα βήματα του αλγορίθμου NRM-SSA και επισημαίνονται οι βασικές διαφορές του από τον FRM-SSA.

*Ενότητα 3:* Προδιαγράφεται η αρχιτεκτονική του συστήματος σε ψηφίδα και παρουσιάζεται η ροή εργασιών που απαιτούνται για την εκτέλεση μιας προσομοίωσης.

*Ενότητα 4:* Αναλύεται η αρχιτεκτονική και η λειτουργικότητα του NRM-SSA SoC, επισημαίνοντας τη διαφοροποίηση των συστημάτων FRM – NRM σε επίπεδο πυρήνα.

*Ενότητα 5:* Αναλύεται συνοπτικά η αρχιτεκτονική και η λειτουργικότητα των μονάδων από το υψηλότερο επίπεδο έως το επίπεδο της υπολογιστικής μονάδας του πυρήνα.

*Ενότητα 6:* Αναλύεται διεξοδικά η αρχιτεκτονική και η λειτουργικότητα των μονάδων που απαρτίζουν την κεντρική μονάδα επεξεργασίας του πυρήνα.

*Ενότητα 7:* Παρουσιάζονται τα στατιστικά αποτελέσματα της σύνθεσης του NRM-SSA SoC και πως αυτά

εξαρτώνται από τις γενικές παραμέτρους του συστήματος. Η ενότητα ολοκληρώνεται με τη θεωρητική ανάλυση των επιδόσεων του συστήματος.

*Ενότητα 8:* Παρουσιάζονται οι επιδόσεις και τα αποτελέσματα πειραμάτων προσομοίωσης βιομοντέλων. Σχολιάζονται τα αποτελέσματα των προσομοιώσεων τόσο σε επίπεδο ορθότητας όσο και σε επίπεδο επιδόσεων σε σύγκριση με γνωστές πλατφόρμες προσομοίωσης.

*Ενότητα 9:* Στην τελευταία ενότητα εκθέτουμε τα συμπεράσματα που προέκυψαν από την παρούσα ερευνητική εργασία, παραθέτουμε τους προβληματισμούς μας και τις πιθανές μελλοντικές μας ενέργειες.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



## 2 ΜΕΘΟΔΟΙ ΣΤΟΧΑΣΤΙΚΗΣ ΠΡΟΣΟΜΟΙΩΣΗΣ

### 2.1 Μοντελοποίηση Βιολογικών Δικτύων

Προκειμένου να προσομοιώσουμε ένα βιολογικό σύστημα πρέπει πρώτα να περιγράψουμε τη συμπεριφορά του σε μορφή δικτύου βιοχημικών αντιδράσεων (βιομοντέλο). Ένα βιομοντέλο απαρτίζεται από  $n$  μοριακά είδη  $\{S_1, \dots, S_n\}$  με αρχικούς πληθυσμούς  $\{X_1, \dots, X_n\}$  που αλληλοεπιδρούν μέσω  $m$  αντιδράσεων  $\{R_1, \dots, R_m\}$ . Προκειμένου να απλοποιηθεί η ανάλυση θεωρούμε ότι όλα τα είδη είναι ομοιόμορφα κατανομημένα μέσα σε έναν όγκο  $\Omega$  στο εσωτερικό ενός κελιού μοναδιαίου μεγέθους. Η υπόθεση αυτή απλοποιεί τους υπολογισμούς, αγνοώντας τις χωρικές επιπτώσεις που υπάρχουν στον πραγματικό κόσμο. Ορίζοντας ως  $X_i(t)$  τη συγκέντρωση του μοριακού είδους  $S_i$  τη χρονική στιγμή  $t$ , τότε η κατάσταση του συστήματος την χρονική στιγμή  $t$  περιγράφεται ως  $X(t) = \{X_1(t), X_2(t), \dots, X_n(t)\}$  με αρχικές συνθήκες  $X_0(t) = x_0$  κατά τη χρονική στιγμή  $t = t_0$ .

Η στοχαστική προσομοίωση ενός βιομοντέλου παρακολουθεί το παραπάνω διάνυσμα καταστάσεων των συγκεντρώσεων του δικτύου σε επιλεγμένα χρονικά διαστήματα, χωρίς τη ρητή επίλυση διαφορικών εξισώσεων από την οποία εξαρτάται η δυναμική του συστήματος. Με βάση την περιγραφή του μοντέλου, όταν μια αντίδραση  $R_\mu$  ενεργοποιηθεί τότε η τρέχουσα κατάσταση  $x$  του συστήματος ενημερώνεται και η νέα κατάσταση είναι η  $X(t + \tau) = x + \nu_\mu$ . Θεωρούμε ότι  $\nu_\mu = (\nu_{1\mu}, \dots, \nu_{n\mu})$ , όπου ο συντελεστής  $\nu_{i\mu}$  αντιπροσωπεύει την αλλαγή της στοιχειομετρίας του είδους  $S_i$  λόγω της ενεργοποίησης της αντίδρασης  $R_\mu$ . Επίσης, κάθε αντίδραση  $R_\mu$  σχετίζεται με ένα σταθερό ρυθμό πιθανότητας  $c_\mu$ , ο οποίος είναι ανάλογος του ρυθμού της αντίδρασης και αντιστρόφως ανάλογος του όγκου  $\Omega$ , όπως παρουσιάζεται στις εξισώσεις 7(α) και 7(β) της αναφοράς [14].

Η πιθανότητα ένας τυχαία επιλεγμένος συνδυασμός αντιδρώντων να αλληλοεπιδράσει και να "πυροδοτήσει" μια αντίδραση  $R_\mu$  στο επόμενο απειροελάχιστο χρονικό διάστημα  $[t, t + dt]$  δίνεται από τον τύπο  $c_\mu * dt$ . Η *τάση*  $a_\mu(x)$  (propensity) μιας αντίδρασης  $R_\mu$  στην κατάσταση  $X$  υπολογίζεται πολλαπλασιάζοντας τη σταθερά  $c_\mu$  με το πλήθος των πιθανών συνδυασμών των μοριακών αντιδρώντων ειδών, όπως φαίνεται στις εξισώσεις (21) έως (26) του [14]. Έτσι, για τη δεύτερης τάξης αντιδράσεις με αντιδρώντα  $S_1$  και  $S_2$ , ισχύει ότι:

$$a_\mu = c_\mu * X_1 * X_2 \quad (1)$$

Η παραπάνω διαδικασία ακολουθεί την Μαρκοβιανή θεωρία, όπου η επόμενη κατάσταση εξαρτάται μόνο από την τρέχουσα. Προσομοιώνοντας κατά αυτόν τον τρόπο το μοντέλο, αποδίδονται οι διαδρομές των καταστάσεων  $X(t)$ . Ο αρχικός αλγόριθμος στοχαστική προσομοίωσης (SSA) του Gillespie, που ονομάζεται Άμεση Μέθοδος (Direct Method, DM) [26], βασίζεται στην παραπάνω διατύπωση και επιταχύνει τη διαδικασία με την εισαγωγή μιας νέας συνάρτησης  $p(\tau, \mu | x, t)$ , η οποία εκφράζει την πιθανότητα η επόμενη αντίδραση  $R_\mu$  να συμβεί μέσα στο επόμενο απειροελάχιστο χρονικό διάστημα  $[t, t + \tau]$ , δεδομένου ότι η τρέχουσα κατάσταση του συστήματος είναι  $X(t) = x$ . Η διαδικασία αυτή πλεονεκτεί, διότι μπορεί να μεταβαίνει από τη μια χρονική στιγμή στην επόμενη χωρίς να χρειάζεται να υπολογιστούν οι ενδιάμεσες καταστάσεις, όπου καμία αντίδραση δεν συμβαίνει. Κατά αυτό τον τρόπο, η ενημέρωση των αντιδρώντων και των προϊόντων συμβαίνει σε διακριτές ποσότητες και η μέθοδος είναι εφαρμόσιμη ακόμα κι αν διατηρούνται σε χαμηλές τιμές οι συγκεντρώσεις των μοριακών ειδών, άλλο ένα αξιοσημείωτο πλεονέκτημα των στοχαστικών σε σχέση με τους

ντετερμινιστικούς αλγορίθμους προσομοίωσης.

Αν και ο DM-SSA είναι αποδοτικός για μοντέλα μικρών βιοχημικών δικτύων, είναι δύσκολο να παραλληλοποιηθεί και απαιτεί πολύ περισσότερο χρόνο για μεσαία και μεγάλα μοντέλα. Λαμβάνοντας υπόψιν αυτούς τους περιορισμούς, ο Gillespie εισήγαγε ως εναλλακτική λύση έναν ισοδύναμο αλγόριθμο στοχαστικής προσομοίωσης, τον First Reaction Method (FRM). Με βάση τον αλγόριθμο αυτόν, υπολογίζεται για κάθε αντίδραση  $R_\mu$  ένας πιθανός (δυναμικός) χρόνος ενεργοποίησης  $\tau_j$ . Το κανάλι αντίδρασης  $R_\mu$  με τον μικρότερο δυναμικό χρόνο  $\tau_\mu$  προσδιορίζεται ως «νικητήρια αντίδραση» (winner reaction), η οποία θα ενεργοποιηθεί στο τέλος του «κύκλου αντιδράσεων» (Reaction Cycle, RC). Δεδομένου ότι ο υπολογισμός του χρόνου  $\tau_j$  για κάθε αντίδραση είναι ανεξάρτητος από τους υπόλοιπους, ο αλγόριθμος επιδέχεται μεγάλο βαθμό παραλληλοποίησης.

Μια ενδιαφέρουσα παρατήρηση που προκύπτει μελετώντας τους αλγορίθμους του Gillespie αφορά την αλληλεξάρτηση των αντιδράσεων που χαρακτηρίζουν ένα βιομοντέλο. Παρατηρούμε ότι κατά την ενεργοποίηση μιας αντίδρασης επηρεάζονται οι συγκεντρώσεις ορισμένων μοριακών ειδών και συνεπώς απαιτείται ο επαναυπολογισμός μόνο των επηρεαζόμενων τάσεων  $a_j(\mathbf{x})$  και αντίστοιχα των χρόνων ενεργοποίησης των επηρεαζόμενων αντιδράσεων. Βασιζόμενοι σε αυτή τη κομβική παρατήρηση οι Gibson και Bruck [17] επινόησαν τον αλγόριθμο στοχαστική προσομοίωσης Next Reaction Method (NRM-SSA), με μικρότερη πολυπλοκότητα εκτέλεσης  $O(\log(m))$  σε σύγκριση με τον αλγόριθμο FRM-SSA που έχει πολυπλοκότητα  $O(m)$ .

## 2.2 Ο Αλγόριθμος NRM των Gibson και Bruck

Στα πλαίσια της Δράσης 1.2 του παρόντος ερευνητικού έργου καλούμαστε να σχεδιάσουμε και να υλοποιήσουμε ένα Σύστημα σε Ψηφίδα (SoC) που να εκτελεί στοχαστική προσομοίωση βιολογικών μοντέλων βασισμένο στον αλγόριθμο των Gibson και Bruck, τον Next Reaction Method (NRM-SSA).

Ακολουθούν τα βήματα του αλγορίθμου NRM-SSA των Gibson και Bruck:

1. Αρχικοποίηση του συστήματος
  - a.  $\mathbf{x} = \mathbf{x}_0$  (καθορισμός των αρχικών πληθυσμών κάθε μοριακού είδους)
  - b.  $t = t_0$  αρχικοποίηση του χρόνου προσομοίωσης
  - c. υπολογισμός του γράφου εξαρτήσεων
  - d. αρχικοποίηση των τάσεων  $a_j(\mathbf{x})$  όλων των αντιδράσεων
  - e. υπολογίζεται για κάθε αντίδραση η χρονική στιγμή ενεργοποίησης της αντίδρασης:

$$\tau_j = \frac{1}{a_j(\mathbf{x})} \ln\left(\frac{1}{r_j}\right) \quad (2)$$

όπου  $r_j$  είναι τυχαίοι αριθμοί στο διάστημα  $[0, 1]$

- f. Αποθήκευση των τάσεων και των χρόνων  $\tau_j$
2. Εύρεση της αντίδρασης  $R_\mu$  με το μικρότερο χρόνο  $\tau_\mu$  ενεργοποίησης χρησιμοποιώντας μηχανισμούς όπως η ουρά προτεραιότητας, δέντρα συγκριτών κ.α.
3. Εκτέλεση της αντίδρασης  $R_\mu$  ανανεώνοντας τη κατάσταση του συστήματος  $\mathbf{x} = \mathbf{x} + \mathbf{v}_\mu$  (μεταβολή των πληθυσμών των μοριακών ειδών βάση του διανύσματος στοιχειομετρίας  $\mathbf{v}_\mu$  της αντίδρασης  $R_\mu$ ) καθώς

και το χρόνο προσομοίωσης  $t = \tau$  όπου  $\tau = \tau_\mu$  (καθολικός χρόνος)

4. Ανανέωση των χρόνων ενεργοποίησης των αντιδράσεων:

- a. Ανανέωση της χρονικής στιγμής ενεργοποίησης των επηρεαζόμενων αντιδράσεων με βάση τον μαθηματικό τύπο:

$$\tau_j = t + \frac{\alpha_{j,old}}{\alpha_j} * (\tau_{j,old} - t) \quad (3)$$

- b. Υπολογισμός της χρονικής στιγμής ενεργοποίησης της αντίδρασης με τον μικρότερο χρόνο ενεργοποίησης με βάση τον μαθηματικό τύπο:

$$\tau_j = t + \frac{1}{\alpha_j(x)} \ln\left(\frac{1}{r_j}\right) \quad (4)$$

όπου  $r_j$  είναι ψευδοτυχαίος αριθμός στο διάστημα (0, 1]

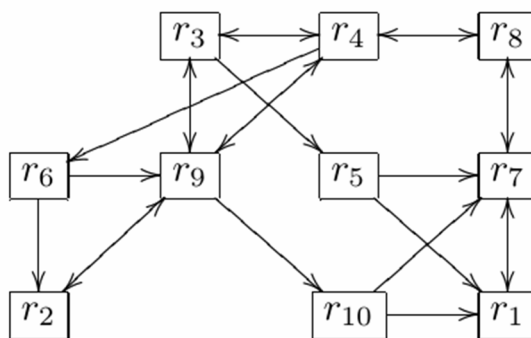
5. Εάν το σύστημα έχει φτάσει τον επιθυμητό χρόνο προσομοίωσης τότε ο αλγόριθμος τερματίζει αλλιώς μεταβαίνει ξανά στο βήμα 2.

### 2.3 Η Διαφοροποίηση του NRM-SSA έναντι του FRM-SSA

Προκειμένου να υλοποιηθεί ο NRM-SSA απαιτούνται τρεις βασικές αλλαγές [27] στον FRM-SSA αλγόριθμο. Η πρώτη αλλαγή αφορά την έννοια του χρόνου. Ο FRM-SSA αλγόριθμος υπολογίζει την σχετική απόσταση του χρόνου ενεργοποίησης της επόμενης αντίδρασης από την τρέχουσα χρονική στιγμή, σε αντίθεση με τον NRM-SSA αλγόριθμο που υπολογίζει την καθολική τιμή του χρόνου ενεργοποίησης όλων των αντιδράσεων. Η δεύτερη διαφοροποίηση σχετίζεται με την έννοια της εξάρτησης των αντιδράσεων. Με βάση τον αλγόριθμο των Gibson και Bruck, απαιτείται η κατασκευή ενός γράφου εξαρτήσεων, όπου για κάθε πιθανή νικητήρια αντίδραση να καθορίζονται οι αντιδράσεις για τις οποίες απαιτείται ο επαναυπολογισμός των τάσεων  $a_j(x)$  (propensity) και αντίστοιχα των χρόνων ενεργοποίησης  $\tau_j$ . Τέλος, η τρίτη αλλαγή αφορά τις δομές δεδομένων που χρησιμοποιεί ο αλγόριθμος NRM έναντι του αλγορίθμου FRM. Δεδομένου ότι οι τάσεις  $a_j(x)$  και οι χρόνοι  $\tau_j$  όλων των αντιδράσεων ενδέχεται να επαναχρησιμοποιηθούν στον επόμενο κύκλο αντιδράσεων, θα πρέπει να αποθηκευτούν σε κατάλληλες δομές δεδομένων ώστε να είναι πάντα διαθέσιμα.

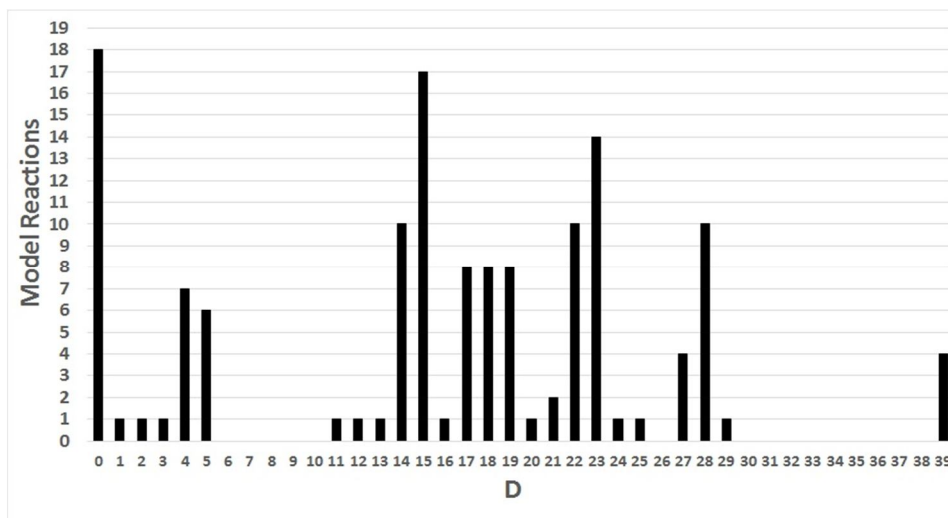
### 2.4 Ο Γράφος Εξαρτήσεων

Η βασική διαφορά ανάμεσα στους αλγορίθμους στοχαστικής προσομοίωσης FRM και NRM είναι ότι ο δεύτερος κάνει χρήση γράφου εξαρτήσεων (Dependency Graph - DG) που απεικονίζει την εξάρτηση μεταξύ των αντιδράσεων. Όπως παρατήρησαν οι Gibson και Bruck, όταν μια αντίδραση συμβεί δεν επηρεάζονται οι συγκεντρώσεις όλων των μοριακών ειδών και συνεπώς δεν απαιτείται ο επαναυπολογισμός των τάσεων  $a_j(x)$  και των χρόνων  $\tau_j$  όλων των αντιδράσεων. Για το λόγο αυτό κατασκευάζεται ένας γράφος εξαρτήσεων, όπου η ακμή από την αντίδραση με τον μικρότερο χρόνο προς άλλες αντιδράσεις υποδηλώνει ότι τουλάχιστον ένα αντιδρόν, ή ένα προϊόν, της «νικητήριας αντίδρασης» συμμετέχει ως αντιδρόν στις αντιδράσεις αυτές, τις λεγόμενες «επηρεαζόμενες αντιδράσεις». Επομένως, σε κάθε κύκλο προσομοίωσης χρησιμοποιούμε τον γράφο εξαρτήσεων ώστε να εξάγουμε τις επηρεαζόμενες αντιδράσεις και να επαναπροσδιορίσουμε τον χρόνο ενεργοποίησης μόνο αυτών των επηρεαζόμενων αντιδράσεων. Στην Εικόνα 1 παρουσιάζεται ένα ενδεικτικό παράδειγμα γράφου εξαρτήσεων για ένα τυχαίο σύνολο αντιδράσεων.



Εικόνα 1: Ενδεικτικό παράδειγμα γράφου εξαρτήσεων αντιδράσεων [27]

Αναλύοντας ένα πραγματικό μοντέλο και σχεδιάζοντας το γράφο εξαρτήσεων μεταξύ των αντιδράσεων, θα διαπιστώσουμε ότι ο γράφος δεν είναι πάντα συμμετρικός καθώς δεν επηρεάζουν όλες οι αντιδράσεις το ίδιο πλήθος αντιδράσεων ( $D$ ). Στην Εικόνα 2, δίδεται ως παράδειγμα το ιστόγραμμα του πλήθους των εξαρτήσεων ενός πραγματικού βιομοντέλου, του ASYN [28]. Το ASYN μοντέλο χαρακτηρίζεται από  $m=136$  αντιδράσεις και  $n=90$  μοριακά είδη και προσομοιώνει τις επιβλαβείς επιδράσεις του πολυμερισμού της πρωτεΐνης *άλφασυνουκλεΐνης* (ASYN) στην ομοιόσταση του κυττάρου, που πιστεύεται ότι συμβάλει στην εμφάνιση της νόσου του Πάρκινσον. Όπως παρουσιάζεται στο ιστόγραμμα της Εικόνας 2, από το σύνολο των αντιδράσεων: 18 αντιδράσεις επηρεάζουν  $D = 0$  αντιδράσεις και 4 αντιδράσεις επηρεάζουν  $D = 39$  αντιδράσεις, ενώ τελικά ο μέσος όρος επηρεαζόμενων αντιδράσεων είναι  $D_{aver} = 15$ .



Εικόνα 2: Το ιστόγραμμα των επηρεαζόμενων αντιδράσεων του ASYN βιομοντέλου [28]. Ο άξονας  $y$  απεικονίζει τον αριθμό των αντιδράσεων του μοντέλου που έχουν το ίδιο πλήθος επηρεαζόμενων αντιδράσεων ( $D$ ).

Παρότι ο NRM-SSA, σε σύγκριση με τον FRM-SSA, κερδίζει σε απόδοση, είναι πολύ δύσκολο να παραλληλοποιηθεί σε επίπεδο υλοποίησης. Υπάρχουν περιπτώσεις όπου μια αντίδραση μπορεί να επηρεάσει όλες τις υπόλοιπες αντιδράσεις ενός βιομοντέλου. Σε αυτή την περίπτωση ο αλγόριθμος NRM-SSA χάνει το πλεονέκτημά του αφού πλέον απαιτούνται  $O(m)$  βήματα εκτέλεσης για την ολοκλήρωση ενός κύκλου αντιδράσεων [27]. Ως εκ τούτου, ο FRM-SSA προτιμάται, στις περισσότερες παράλληλες υλοποιήσεις

στοχαστικής προσομοίωσης, ενώ ο NRM-SSA προτιμάται για μοντέλα με πάρα πολλές αντιδράσεις και μικρό σχετικά μέσο όρο επηρεαζόμενων αντιδράσεων ( $D_{aver}$ ).

Η διαθεσιμότητα εναλλακτικών υλοποιήσεων και για τους δύο αλγορίθμους διευκολύνει τη βέλτιστη επιλογή που εξαρτάται από το πλήθος των αντιδράσεων του μοντέλου ( $m$ ) και από τα χαρακτηριστικά του ιστογράμματος εξαρτήσεων του μοντέλου ( $D_{aver}$ ). Εν τέλει, στην ιδανική περίπτωση, ένας προσομοιωτής θα πρέπει να επιλέξει τον κατάλληλο αλγόριθμο προσομοίωσης με βάση την πολυπλοκότητα του μοντέλου και το ιστογράμμο εξαρτήσεων, ώστε να μεγιστοποιείται η αποδοτικότητα του συστήματος στοχαστικής προσομοίωσης με ευέλικτο τρόπο.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
Επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
Πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



### 3 ΠΡΟΔΙΑΓΡΑΦΕΣ ΣΥΣΤΗΜΑΤΟΣ

Στην ενότητα αυτή θα παρουσιαστούν οι προδιαγραφές και ο τρόπος λειτουργίας του παραμετρικού συστήματος NRM-SSA SoC που σχεδιάστηκε και αναπτύχθηκε. Ο τρόπος λειτουργίας είναι παρόμοιος με αυτόν που περιγράφεται στο Παραδοτέο 1.1 για το παραμετρικό FRM-SSA SoC, αλλά δεδομένου ότι είναι τελείως διαφορετικός ο αλγόριθμος που υλοποιείται, παρουσιάζονται και σημαντικές διαφορές στις βασικές παραμέτρους τις οποίες θα περιγράψουμε στη συνέχεια.

Το σύστημα απαρτίζεται από έναν γενικού σκοπού ηλεκτρονικό υπολογιστή (Host-PC) που αφενός φιλοξενεί τα δεδομένα του μοντέλου που θα προσομοιωθεί, και αφετέρου συνδέεται με αναπτυξιακή κάρτα που ενσωματώνει ολοκληρωμένο κύκλωμα επαναπρογραμματιζόμενης λογικής (Field Programmable Gate Array, FPGA). Στην περίπτωση μας η αναπτυξιακή κάρτα είναι η KC705 της εταιρίας Xilinx [29], που ενσωματώνει ένα μεσαίου μεγέθους FPGA, το XC7K325-2FFG900C [30], με 203,800 LUTs, 445 BRAMs (36Kbit) και 840 DSPs. Για την ανάπτυξη του συστήματος χρησιμοποιήθηκαν εργαλεία της ίδιας εταιρίας και ο σχεδιασμός πραγματοποιήθηκε χρησιμοποιώντας τη γλώσσα περιγραφής υλικού VHDL. Τέλος, για την επικοινωνία του Host-PC με το NRM-SSA SoC χρησιμοποιείται το πρωτόκολλο επικοινωνίας PCIe.

#### 3.1 Προδιαγραφές της παραμετρικής αρχιτεκτονικής NRM SoC

Στο Παραδοτέο 1.1 έχει γίνει εκτενής ανάλυση του συστήματος που ενσωματώνει το FRM-SSA SoC. Εγινε προσπάθεια το σύστημα αυτό να επαναχρησιμοποιηθεί κατά το δυνατόν με στόχο τη μείωση του χρόνου σχεδίασης του NRM-SSA SoC. Τα δύο συστήματα παρουσιάζουν μικρές αλλαγές στο επίπεδο του πυρήνα (SSA Core), αλλά καθώς εξετάζουμε τα χαμηλότερα επίπεδα σχεδίασής τους οι διαφορές πληθαίνουν. Στη συνέχεια της ενότητας αυτής γίνεται εκτενής αναφορά στην αρχιτεκτονική του πυρήνα NRM-SSA, από το υψηλότερο προς το χαμηλότερο επίπεδο σχεδίασης, επισημαίνοντας τις διαφορές του από τον FRM-SSA Core.

Ο πυρήνας NRM-SSA χαρακτηρίζεται από γενικές παραμέτρους που μπορούν ευέλικτα να καθορίζουν τη δομή του, τη συμπεριφορά του και το μέγεθος των βιομοντέλων που μπορεί να εκτελέσει με βάση τις παραμέτρους προσομοίωσης του χρήστη. Η δυνατότητα της παραμετροποίησης του πυρήνα κατά το στάδιο σύνθεσης τον κατατάσσει στους «μαλακούς πυρήνες» (Soft IP Core) υλικού ειδικού σκοπού. Το συνολικό σύστημα NRM-SSA SoC είναι μια πολυπύρηνη αρχιτεκτονική, 32-bit, πλήρους διοχέτευσης (fully pipelined) με τη δυνατότητα να εμπεριέχονται ένας ή περισσότεροι διασυνδεδεμένοι πυρήνες στο ανώτερο επίπεδο.

Όπως προαναφέρθηκε το σύστημα είναι παραμετρικό σύμφωνα με τα χαρακτηριστικά ενός βιολογικού δικτύου. Οι παράμετροι δίνονται ως generic μεταβλητές στο υψηλότερο αρχείο (top-level) της υλοποίησης του SoC και κληρονομούνται σε κάθε επίπεδο του σχεδίου. Στη συνέχεια ακολουθούν τα χαρακτηριστικά που παραμετροποιούν την αρχιτεκτονική NRM SoC:

- Αριθμός μοριακών ειδών –  $n$ . Η παράμετρος αυτή καθορίζει το βάθος των μνημών όπου αποθηκεύονται τα μοριακά είδη και αντίστοιχα το μέγιστο επιτρεπόμενο πλήθος διαφορετικών μοριακών ειδών ενός μοντέλου.
- Αριθμός αντιδράσεων –  $m$ . Η παράμετρος αυτή καθορίζει το βάθος της μνήμης όπου αποθηκεύονται οι αντιδράσεις του δικτύου και αντίστοιχα το μέγιστο επιτρεπόμενο πλήθος των αντιδράσεων ενός μοντέλου. Οι αντιδράσεις μπορεί να είναι μέχρι και  $3^{00}$  βαθμού με 5 προϊόντα.
- Μέσος όρος επηρεαζόμενων αντιδράσεων –  $D_{aver}$ . Η παράμετρος αυτή καθορίζει το πλάτος της μνήμης όπου αποθηκεύεται ο γράφος εξαρτήσεων και αντιστοιχεί στο μέγιστο επιτρεπόμενο μέσο όρο επηρεαζόμενων αντιδράσεων ενός μοντέλου.

- Αριθμός NRM πυρήνων –  $C$ . Η παράμετρος  $C$  (COREs) καθορίζει τον αριθμό των παράλληλων NRM πυρήνων που θα διαθέτει το σύστημα μετά τη σύνθεση, όπου κάθε πυρήνας μπορεί να εκτελεί ανεξάρτητες προσομοιώσεις ενός βιομοντέλου.

Κάθε μία από τις παραπάνω παραμέτρους καθορίζει τον τρόπο λειτουργίας του συστήματος, τόσο σε επίπεδο προσομοίωσης όσο και σε επίπεδο αρχιτεκτονικής. Αφενός οι παράμετροι  $m$ ,  $n$  και  $D_{aver}$  καθορίζουν τις απαιτήσεις ενός πυρήνα σε υπολογιστικούς πόρους και επιφάνεια με βάση τα χαρακτηριστικά μεγέθη ενός μοντέλου, και αφεταίρου η παράμετρος  $C$  καθορίζει πόσοι πυρήνες θα εμπεριέχονται στο σύστημα μετά τη σύνθεση. Το πλήθος των πυρήνων εξαρτάται από το τελικό μέγεθος του πυρήνα με βάση τις υπόλοιπες παραμέτρους αλλά και από τους διαθέσιμους πόρους του επιλεγμένου FPGA. Το σύστημα έχει σχεδιαστεί ώστε να μπορεί να επεκταθεί χωρίς περιορισμούς ανάλογα με το μέγεθος του FPGA που διαθέτει ο χρήστης.

### 3.2 Λειτουργία Συστήματος και Ροή Εργασιών

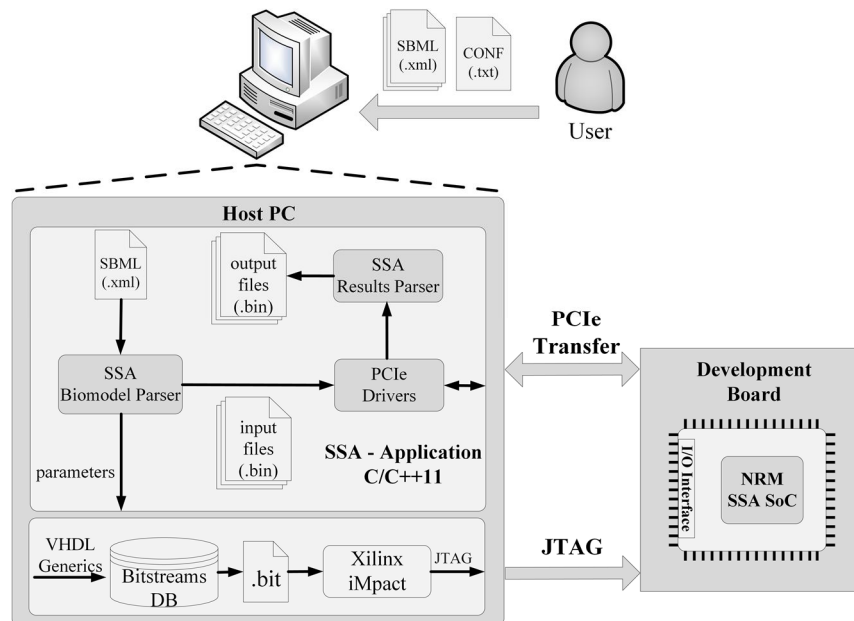
Το Host-PC είναι υπεύθυνο για την ομαλή διεξαγωγή της προσομοίωσης και, συνοπτικά, η εκτέλεση ενός βιομοντέλου ακολουθεί την εξής διαδικασία: αρχικά, με τη χρήση κατάλληλα υλοποιημένου λογισμικού μεταφέρονται στο NRM-SSA SoC τα δεδομένα του μοντέλου και οι επιθυμητές παράμετροι που ορίζει ο χρήστης. Δίνεται από το Host-PC το σήμα εκκίνησης της προσομοίωσης. Καθ' όλη τη διάρκεια της προσομοίωσης, το NRM-SSA SoC παράγει πληροφορίες οι οποίες συλλέγονται και αποθηκεύονται στο Host-PC. Με βάση τις πληροφορίες αυτές, παράγονται ένα ή περισσότερα αρχεία με τα αποτελέσματα και τα στατιστικά της προσομοίωσης. Ο χρήστης έχει τη δυνατότητα να επισκοπήσει τα αποτελέσματα, να αποθηκεύσει την επιθυμητή πληροφορία και εν τέλει να εκτελέσει μια νέα προσομοίωση με νέες παραμέτρους, δεδομένα ή αρχικές τιμές.

Η παραπάνω διαδικασία χωρίζεται σε 3 φάσεις και απεικονίζεται συνοπτικά στην Εικόνα 3. Κατά την πρώτη φάση προσδιορίζονται οι παράμετροι του συστήματος που απαιτείται και τίθεται σε λειτουργία το σύστημα. Στη δεύτερη φάση πραγματοποιείται η προσομοίωση του δοθέντος μοντέλου με βάση τις παραμέτρους που επιθυμεί ο χρήστης. Και τέλος, στη τρίτη φάση αναλύονται τα δεδομένα που προέκυψαν.

Στη πρώτη φάση ο χρήστης διαμορφώνει τη μορφή της αρχιτεκτονικής μέσω των γενικών παραμέτρων στο υψηλότερο (top-level) αρχείο σχεδίασης VHDL. Αφού καθορίσει τις γενικές παραμέτρους της αρχιτεκτονικής μέσω του λογισμικού ISE [31] παράγει το δυαδικό αρχείο bitstream [32] - το οποίο περιέχει τη περιγραφή της αρχιτεκτονικής σε επίπεδο κυκλώματος. Έπειτα, αφού παραχθεί το bitstream ο χρήστης μέσω του λογισμικού iMpract [33] προγραμματίζει το FPGA μεταφέροντας το bitstream μέσω του πρωτοκόλλου επικοινωνίας JTAG [34]. Εδώ θα πρέπει να τονιστεί ότι σκοπός της παρούσας έρευνας είναι τελικά να δημιουργηθεί μια βιβλιοθήκη από bitstreams προκειμένου το βήμα της παραγωγής του bitstream (πρώτη φάση) να αποφεύγεται καθώς αποτελεί χρονοβόρα διαδικασία. Στόχος είναι να απαιτείται να δημιουργηθεί ένα νέο bitstream μόνο σε εξαιρετικές περιπτώσεις όπου οι ανάγκες κάποιου χρήστη εξειδικεύονται τόσο που επιβάλουν το βήμα αυτό. Στις περισσότερες περιπτώσεις οι παράμετροι του χρήστη χρησιμοποιούνται για την επιλογή του πλέον κατάλληλου bitstream (SoC instance) από αυτά που έχουν ήδη δημιουργηθεί εκ των προτέρων αξιοποιώντας τη παραμετρική φύση των SoC περιγραφών.

Με την ολοκλήρωση της φάσης του προγραμματισμού του FPGA ο χρήστης μπορεί να προχωρήσει στη δεύτερη φάση που περιλαμβάνει τον προγραμματισμό του NRM SoC συστήματος με το επιθυμητό βιομοντέλο και την εκτέλεση του in-silico πειράματος. Σε αυτή τη φάση ο χρήστης αποθηκεύει στο Host-PC το βιομοντέλο

που επιθυμεί να προσομοιώσει σε standard SBML κωδικοποίηση [35]. Το SBML αρχείο περιέχει το σύνολο αντιδράσεων που περιγράφουν το βιολογικό μοντέλο καθώς και τα μοριακά είδη και τους αρχικούς πληθυσμούς τους. Τα στοιχεία αυτά πρέπει να μετατραπούν σε κατάλληλες δομές δεδομένων που αναγνωρίζει το NRM SoC σε δυαδική μορφή.



Εικόνα 3: Ο χρήστης δίνοντας ως είσοδο το SBML [35] αρχείο του βιομοντέλου και το αρχείο παραμέτρων, θα λάβει ως έξοδο τα αποτελέσματα της επιθυμητής προσομοίωσης.

Τη μετατροπή αυτή την αναλαμβάνει ένα εργαλείο λογισμικού που αναπτύχθηκε στα πλαίσια του έργου και ονομάζεται SBML Biomodel Parser. Εκτός από τις αντιδράσεις του βιομοντέλου και τα μοριακά είδη θα πρέπει να ρυθμιστούν και οι παράμετροι που σχετίζονται με την εκτέλεση της προσομοίωσης του βιομοντέλου. Αυτές οι παράμετροι ρυθμίζονται από τον χρήστη μέσω του Αρχείου Ρυθμίσεων (configuration file) και είναι οι εξής:

1.  $T_{sim}$ : Χρόνος ολοκλήρωσης της προσομοίωσης σε δευτερόλεπτα
2.  $T_{sam}$ : Περίοδος δειγματοληψίας των αποτελεσμάτων σε δευτερόλεπτα.
3. **Στοχαστικές Επαναλήψεις Προσομοίωσης (Repetitions - R)**: αριθμός των στοχαστικών επαναλήψεων που θα εκτελεστεί το πείραμα.
4. **Μέγιστος Αριθμός Εσωτερικών Βημάτων (Max Internal Steps)**: μέγιστος αριθμό κύκλων αντιδράσεων (reaction cycles) που επιτρέπεται να μεσολαβήσουν κατά τη διάρκεια μιας περιόδου δειγματοληψίας προτού τερματιστεί μια επανάληψη του πειράματος.

Αφού συμπληρωθεί και το Αρχείο Ρυθμίσεων, τροφοδοτείται μαζί με το SBML αρχείο στο εργαλείο SBML Biomodel Parser προκειμένου να παραχθούν τα κατάλληλα δυαδικά αρχεία. Όταν οριστικοποιηθούν όλα τα απαραίτητα αρχεία δίδεται το σήμα εκκίνησης της προσομοίωσης. Καθ' όλη τη διάρκεια της προσομοίωσης το NRM-SSA SoC παράγει σε κάθε περίοδο δειγματοληψίας αποτελέσματα (πληθυσμοί μοριακών ειδών) τα οποία αποστέλλονται στο Host-PC και αποθηκεύονται σε ένα αρχείο αποτελεσμάτων (results.bin).

Με την ολοκλήρωση της προσομοίωσης ξεκινάει η τρίτη φάση του συστήματος η οποία περιλαμβάνει την ανάλυση των αποτελεσμάτων. Τα αποτελέσματα που παράγονται από τους NRM πυρήνες μεταφέρονται μέσω του PCIe πρωτοκόλλου στο Host PC όπου και αποθηκεύονται σε ένα δυαδικό αρχείο (results.bin). Με το τέλος



της προσομοίωσης το αρχείο results.bin τροφοδοτείται σε ένα άλλο εργαλείο λογισμικού που αναπτύχθηκε στα πλαίσια του έργου και είναι υπεύθυνο για την ανάλυση των δεδομένων. Το εργαλείο, SSA Results Parser, αρχικά ξεχωρίζει τα δεδομένα που παρήγαγε ο κάθε NRM πυρήνας καθότι είναι δυνατόν ο κάθε πυρήνας να εκτελεί ένα τελείως διαφορετικό πείραμα. Έπειτα για κάθε πείραμα που διεξήχθη, το εργαλείο SSA Results Parser υπολογίζει μέσες τιμές, ελάχιστα και μέγιστα για όλους τους μοριακούς πληθυσμούς, εφόσον έχουν εκτελεστεί περισσότερες από μια στοχαστικές επαναλήψεις. Τα στατιστικά αυτά μαζί με τα αποτελέσματα κάθε επανάληψης αποθηκεύονται σε CSV αρχεία [36] προκειμένου να δίνεται η δυνατότητα στον χρήστη να τα αναλύσει περαιτέρω με άλλα λογισμικά. Επιπλέον το εργαλείο δημιουργεί CSV αρχεία με στατιστικά που σχετίζονται με τη προσομοίωση όπως: ο συνολικός αριθμός κύκλων προσομοίωσης που εκτελέστηκαν, ο συνολικός αριθμός περιόδων δειγματοληψίας που καταγράφηκαν καθώς και ο χρόνος που χρειάστηκε για να ολοκληρωθεί η κάθε επανάληψη.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*  
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

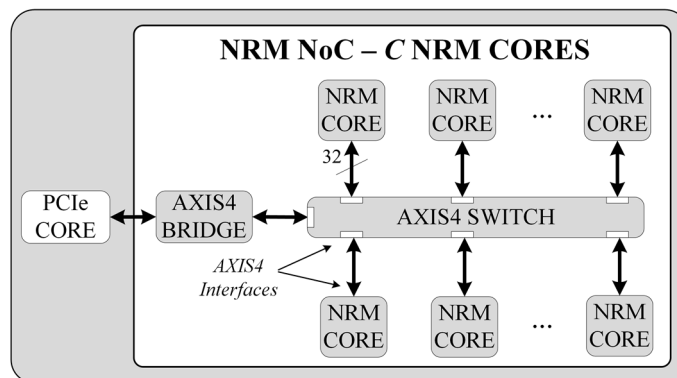


ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

## 4 ΑΡΧΙΤΕΚΤΟΝΙΚΗ NRM ΣΥΣΤΗΜΑΤΟΣ ΣΕ ΨΗΦΙΔΑ

Όπως έχει αναφερθεί και στα προηγούμενα κεφάλαια, στόχος μας είναι η αρχιτεκτονική του NRM SoC να είναι ευέλικτη και όσο το δυνατόν περισσότερο παραμετρική. Οι γενικές παράμετροι (generic parameters) που σχετίζονται με την αρχιτεκτονική του NRM SoC αφορούν τόσο τα υποστηριζόμενα μοντέλα βιολογικών δικτύων όσο και τη δομή του συστήματος που ενσωματώνει έναν ή περισσότερους πυρήνες.

Το NRM SoC είναι σχεδιασμένο να περιέχει περισσότερους από έναν πυρήνες και για το λόγο αυτό ορίζεται η γενική παράμετρος  $C$  (COREs) ώστε να καθορίζεται, πριν τη σύνθεση, το πλήθος των πυρήνων NRM που θα εμπεριέχονται στο NRM SoC. Η τιμή της παραμέτρου εξαρτάται από τις υπόλοιπες παραμέτρους του συστήματος αλλά και από τους διαθέσιμους πόρους του FPGA (LUTs, FFs, BRAMs, DSPs).



Εικόνα 4: Η γενική αρχιτεκτονική δομή του NRM SoC. Το πλήθος των πυρήνων είναι παραμέτρος του συστήματος.

Όπως φαίνεται στην Εικόνα 4, το NRM-SSA SoC είναι δυνατόν να περιέχει περισσότερους από έναν πυρήνες. Κάθε πυρήνας είναι ανεξάρτητος από τους υπόλοιπους και όλοι μαζί συνθέτουν ένα δίκτυο τοπολογίας αστέρα. Για να γίνει επιτυχής η επικοινωνία των πυρήνων με το Host-PC, κάθε πυρήνας συνδέεται στη θύρα ενός διανομέα (Switch) και χρησιμοποιούν ένα πρωτόκολλο επικοινωνίας που ακολουθεί πλήρως τις προδιαγραφές του AXI4 Stream Interface – AXIS4 του AMBA Bus [37] [38].

### 4.1 Αρχιτεκτονική Πυρήνα NRM-SSA

Ένας πυρήνας NRM-SSA είναι σε θέση να προσομοιώσει ένα οποιοδήποτε μοντέλο που έχει  $m$  αντιδράσεις ( $m$  στο διάστημα  $[2, 4096]$ ) και  $n$  μοριακά είδη ( $n$  στο διάστημα  $[2, 4096]$ ), αναλόγως τις γενικές παραμέτρους (generic parameters) που έχουν οριστεί κατά τη σύνθεση του πυρήνα. Αν και η υλοποίηση εφαρμόζει ακριβώς τα βήματα του αλγορίθμου Next Reaction Method (NRM), στο επίπεδο του πυρήνα η ροή των δεδομένων δεν διαφέρει σχεδόν καθόλου από τη ροή των δεδομένων ενός πυρήνα FRM-SSA.

Η γενική ιδέα ήταν να επαναχρησιμοποιηθούν κατά το δυνατόν οι μονάδες που είχαν σχεδιαστεί για τον πυρήνα FRM και να επανασχεδιαστούν μόνο τα σημεία στα οποία διαφέρουν οι δύο αλγόριθμοι. Συνοπτικά η λειτουργία των βασικών μονάδων του πυρήνα NRM-SSA είναι η εξής:

- Input Interface System – IIS: υπεύθυνη για τα εισερχόμενα δεδομένα του πυρήνα από το δίκτυο.
- Output Interface System – OIS: υπεύθυνη για τα εξερχόμενα δεδομένα του πυρήνα προς το δίκτυο.

- Simulation Control Unit – SCU: υπεύθυνη για τον έλεγχο και την εξέλιξη της προσομοίωσης που εκτελείται στο πυρήνα.
- ST Initialization Memory System – STI MS: υπεύθυνη για τη αποθήκευση των αρχικών τιμών των πληθυσμών των μοριακών ειδών με σκοπό την αρχικοποίηση του συστήματος στην αρχή κάθε επανάλληψης της προσομοίωσης.
- Species Update Unit – SPU: υπεύθυνη για την ανανέωση των μοριακών πληθυσμών με βάση τη νικητήρια αντίδραση που έχει προκύψει στο τέλος κάθε κύκλου αντιδράσεων.
- NRM Processing Unit – NPU: είναι η καρδιά του NRM πυρήνα και εκτελεί τις βασικές λειτουργίες του NRM αλγορίθμου (μαθηματικές πράξεις και συγκρίσεις). Αν και η είσοδος/έξοδος αυτής της οντότητας παραμένει η ίδια, η εσωτερική λειτουργία διαφέρει αρκετά από την αντίστοιχη μονάδα FPU του πυρήνα FRM-SSA.

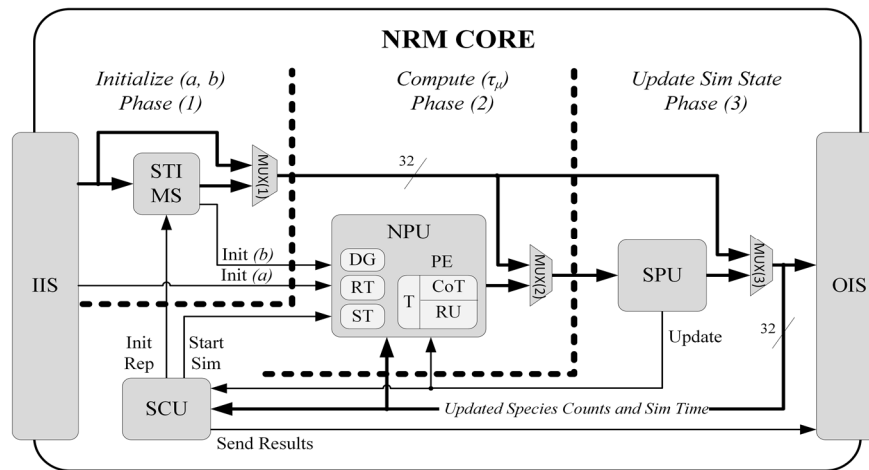
Ακολουθεί ο πίνακας όπου εμπεριέχονται όλα τα σήματα εισόδου και εξόδου της μονάδας NRM Core, καθώς και οι γενικές παράμετροι που καθορίζονται πριν τη σύνθεση του πυρήνα.

Πίνακας 1: Όλα τα σήματα εισόδου και εξόδου του NRM Core, καθώς και οι γενικές παράμετροι του πυρήνα.

Όνομα Σήματος	Περιγραφή
Γενικές Παράμετροι (Generic Parameters)	
RT_SIZE	Το μέγιστο πλήθος αντιδράσεων ( $m$ ) που μπορεί να υποστηρίξει ο πυρήνας.
ST_SIZE	Το μέγιστο πλήθος μοριακών ειδών ( $n$ ) που μπορεί να υποστηρίξει ο πυρήνας.
DG_SIZE	Ο μέγιστος μέσος όρος των επηρεαζόμενων αντιδράσεων που μπορεί να υποστηρίξει ο πυρήνας ( $\max D_{aver}$ ).
H	Το ύψος του δέντρου συγκριτών στην υπολογιστική μονάδα του πυρήνα.
Είσοδος	
s_axis_tid	Όλα τα απαραίτητα σήματα εισόδου με βάση το AXI4 Stream Interface – AXIS4 που εδραιώνουν την επικοινωνία του πυρήνα με τον διανομέα (Switch).
s_axis_tvalid	
s_axis_tready	
s_axis_tdata	
Έξοδος	
m_axis_status	Όλα τα απαραίτητα σήματα εξόδου με βάση το AXI4 Stream Interface – AXIS4 που εδραιώνουν την επικοινωνία του πυρήνα με τον διανομέα (Switch).
m_axis_tready	
m_axis_tvalid	
m_axis_tdata	

## 4.2 Λειτουργία Πυρήνα και Ροή Δεδομένων

Η Εικόνα 5 απεικονίζει την δομή ενός πυρήνα NRM-SSA. Στη συνέχεια παρουσιάζεται ο τρόπος λειτουργίας του και αναλύονται οι σχεδιαστικές επιλογές ως προς την διαφοροποίηση του NRM αλγορίθμου.



Εικόνα 5: Η αρχιτεκτονική ενός πυρήνα NRM-SSA

Η λειτουργία του πυρήνα NRM χωρίζεται σε τρεις φάσεις και είναι οι εξής:

#### Φάση 1 - Αρχικοποίηση:

Η φάση αρχικοποίησης αποτελείται από δύο στάδια. Το πρώτο στάδιο (*Initialize 1.a*) εκτελείται μια φορά κατά την εκκίνηση της προσομοίωσης και περιλαμβάνει την αρχικοποίηση της δομής STI καθώς και των εσωτερικών δομών της μονάδας NPU (RT, DG, RNG). Η SCU είναι υπεύθυνη να προωθήσει τα δεδομένα από την είσοδο του πυρήνα στην είσοδο της εκάστοτε δομής ενεργοποιώντας το σήμα *Init(a)*. Το δεύτερο στάδιο (*Initialize 1.b*) εκτελείται στην αρχή κάθε επανάληψης της προσομοίωσης και περιλαμβάνει την αρχικοποίηση της δομής ST μέσα στη μονάδα NPU. Η SCU ενεργοποιεί το σήμα *Init\_Rep*, ώστε να προωθηθούν τα δεδομένα από τη δομή STI στη δομή ST, ενεργοποιώντας το σήμα *Init(b)*.

#### Φάση 2 – Εύρεση της Επόμενης Νικητήριας Αντίδρασης:

Αφού ολοκληρωθεί η αρχικοποίηση, η SCU ενεργοποιεί το σήμα *Start\_Sim* και ένας κύκλος αντιδράσεων ξεκινάει. Στη φάση αυτή, η μονάδα NPU εκτελεί όλες τις μαθηματικές και λογικές πράξεις που απαιτούνται ώστε να βρεθεί η επόμενη "νικητήρια αντίδραση". Η SCU θα ενεργοποιήσει την συγκεκριμένη φάση εάν και μόνο αν ο τρέχον χρόνος προσομοίωσης είναι μικρότερος από τον μέγιστο επιθυμητό που έχει ορίσει ο χρήστης ( $t_{sim} < T_{sim}$ ).

#### Φάση 3 – Ανανέωση πληθυσμών

Όταν ολοκληρωθεί η Φάση 2, έχει βρεθεί η επόμενη νικητήρια αντίδραση, η αντίδραση δηλαδή με τον μικρότερο δυνητικά χρόνο ενεργοποίησης. Κατά τη Φάση 3, η μονάδα SPU εκτελεί τις πράξεις ανανέωσης των πληθυσμών των στοιχείων που συμμετέχουν στη νικητήρια αντίδραση, ενώ η SCU ανανεώνει τον τρέχοντα χρόνο προσομοίωσης και πραγματοποιεί τους εξής ελέγχους:

- Αν στη συγκεκριμένη χρονική στιγμή πρέπει να πραγματοποιηθεί δειγματοληψία τότε ενεργοποιείται ο μηχανισμός που προωθεί τα δεδομένα από την SPU στο Host-PC με τη βοήθεια της μονάδας OIS. Ο πυρήνας μεταβαίνει στη Φάση 2.
- Αν έχει παρέλθει ο επιθυμητός χρόνος προσομοίωσης, η προσομοίωση σταματάει και ο πυρήνας μεταβαίνει στη Φάση 1, ώστε να γίνει η αρχικοποίηση της δομής ST (*Initialize 1.b*) και να ξεκινήσει η

επόμενη επανάληψη της προσομοίωσης (*Init\_Rep*).

- Αν ο πυρήνας έχει εκτελέσει όλο το πλήθος των επαναλήψεων θα πρέπει να τερματίσει ολοκληρωτικά την προσομοίωση και να ενημερώσει το Host-PC.
- Διαφορετικά, συνεχίζεται η προσομοίωση μεταβαίνοντας στην Φάση 2.

Αν και η παραπάνω ροή δεδομένων είναι σταθερή, τόσο στον πυρήνα FRM-SSA όσο και στον πυρήνα NRM-SSA, παρόλα αυτά, τα δύο συστήματα έχουν και σημαντικές διαφορές που απαιτούν τον επανασχεδιασμό συγκεκριμένων μονάδων.

### 4.3 Διαφοροποίηση Πυρήνων FRM - NRM

Σε αυτό το επίπεδο σχεδίασης οι διαφορές που παρουσιάζουν οι πυρήνες NRM και FRM είναι:

- Προστέθηκε μια κατάσταση αρχικοποίησης του γράφου εξαρτήσεων (Dependency Graph) μέσα στη μονάδα ελέγχου της προσομοίωσης του πυρήνα (SCU).
- Διαφέρει ο τρόπος διαχείρισης του χρόνου προσομοίωσης.

Το σύστημα για να αρχίσει να εκτελεί την προσομοίωση του μοντέλου, θα πρέπει πρώτα να αρχικοποιηθούν όλες οι παράμετροι, όλες οι δομές δεδομένων και όλα τα σήματα ελέγχου των μονάδων του πυρήνα NRM. Για τις ανάγκες του αλγορίθμου, μέσα στην επεξεργαστική μονάδα του πυρήνα (NRM Processing Unit, NPU), έχει σχεδιαστεί μια επιπλέον δομή δεδομένων που αποθηκεύει τον γράφο εξαρτήσεων στη μορφή πίνακα. Για τον λόγο αυτόν, μέσα στη μονάδα ελέγχου της προσομοίωσης του πυρήνα (SCU) έχει προστεθεί μια νέα κατάσταση που φροντίζει να αρχικοποιηθεί κατάλληλα ο πίνακας αυτός. Περισσότερες λεπτομέρειες για το γράφο εξαρτήσεων δίνονται σε επόμενη ενότητα που αναλύει το επίπεδο της υπολογιστικής μονάδας του NRM Core, την NPU (NRM Processing Unit).

Η δεύτερη αλλαγή επηρεάζει τη διαχείριση του χρόνου προσομοίωσης. Όπως έχει αναφερθεί και προηγουμένως, ο FRM αλγόριθμος υπολογίζει τη σχετική απόσταση του χρόνου της επόμενης αντίδρασης από την τρέχουσα χρονική στιγμή, ενώ ο NRM αλγόριθμος υπολογίζει την καθολική τιμή του χρόνου που θα εκτελεστεί η επόμενη αντίδραση. Επομένως, κατά τον FRM αλγόριθμο το αποτέλεσμα ενός κύκλου αντίδρασης είναι ο χρόνος που θα προστεθεί στον χρόνο προσομοίωσης, ενώ κατά τον NRM αλγόριθμο το αποτέλεσμα είναι ο ίδιος ο χρόνος προσομοίωσης. Και στις δύο περιπτώσεις η αντίδραση με τον μικρότερο χρόνο είναι η νικητήρια, αλλά στην περίπτωση του NRM ο χρόνος κάθε αντίδρασης αποθηκεύεται σε μια δομή δεδομένων και επαναχρησιμοποιείται στον επόμενο κύκλο αντίδρασης.

Συνεπώς, στο τέλος κάθε κύκλου αντιδράσεων η NPU (NRM Processing Unit) προωθεί τα στοιχεία της νικητήριας αντίδρασης στην μονάδα SPU (Species Update Unit) ώστε να ενημερωθεί κατάλληλα ο πληθυσμός των μοριακών ειδών του μοντέλου σύμφωνα με την νικητήρια αντίδραση, και στη μονάδα SCU (Simulation Control Unit) η οποία είναι υπεύθυνη να αποθηκεύσει τον χρόνο της νικητήριας αντίδρασης, να ελέγξει αν πρέπει να γίνει δειγματοληψία σύμφωνα με την περίοδο δειγματοληψίας ( $T_{sam}$ ), να τερματιστεί ή να συνεχιστεί η προσομοίωση αναλόγως αν η προσομοίωση έχει φτάσει στον μέγιστο επιθυμητό χρόνο προσομοίωσης ( $T_{sim}$ ) που έχει ορίσει ο χρήστης ή όχι.

Αν και οι περισσότερες μονάδες του πυρήνα (NRM Core) παραμένουν ίδιες ή τροποποιούνται ελαφρώς, η υπολογιστική μονάδα NPU (NRM Processing Unit) διαφέρει αισθητά από την αντίστοιχη υπολογιστική μονάδα FRM (FRM Processing Unit). Η επόμενη ενότητα παρουσιάζει αναλυτικά την αρχιτεκτονική της και ποιες είναι η σχεδιαστικές αλλαγές που εφαρμόστηκαν.

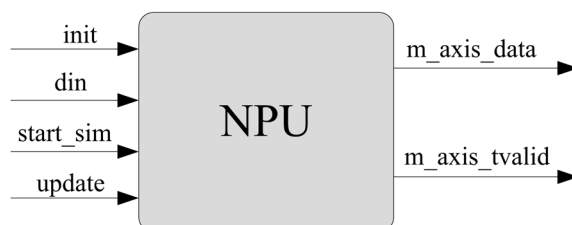


## 5 ΑΡΧΙΤΕΚΤΟΝΙΚΗ NRM ΕΠΕΞΕΡΓΑΣΤΙΚΗΣ ΜΟΝΑΔΑΣ

Μια από τις ιδιαιτερότητες του NRM Core ως προς τη σχεδίαση για υλοποίηση σε FPGA είναι η μη παραλληλοποιήσιμη δομή του αλγορίθμου και κατ' επέκταση του σχεδίου του NRM πυρήνα. Ο FRM αλγόριθμος απαιτεί τον υπολογισμό των χρόνων  $t_j$  όλων των αντιδράσεων, μια διαδικασία η οποία μπορεί να ανατεθεί σε περισσότερες από μια παράλληλες υπολογιστικές μονάδες (Processing Element – PE). Ο NRM αλγόριθμος, όμως, χειρίζεται πολύ μικρότερο αριθμό αντιδράσεων σε κάθε κύκλο αντίδρασης και συνεπώς δεν απαιτούνται παράλληλοι υπολογισμοί.

### 5.1 NPM Processing Unit - NPU

Η μονάδα NRM Processing Unit (NPU) είναι η υπολογιστική μονάδα του πυρήνα NRM και η αρχιτεκτονική της είναι διαφορετική από την αντίστοιχη FPU του πυρήνα FRM (Παραδοτέο 1.1). Οι διαφορές επικεντρώνονται κυρίως στην εισαγωγή της δομής του γράφου εξαρτήσεων, στη δομή της υπολογιστικής μηχανής και στα διάφορα σήματα ελέγχου που προστίθενται για τον έλεγχο των εσωτερικών μονάδων της μονάδας NPU.

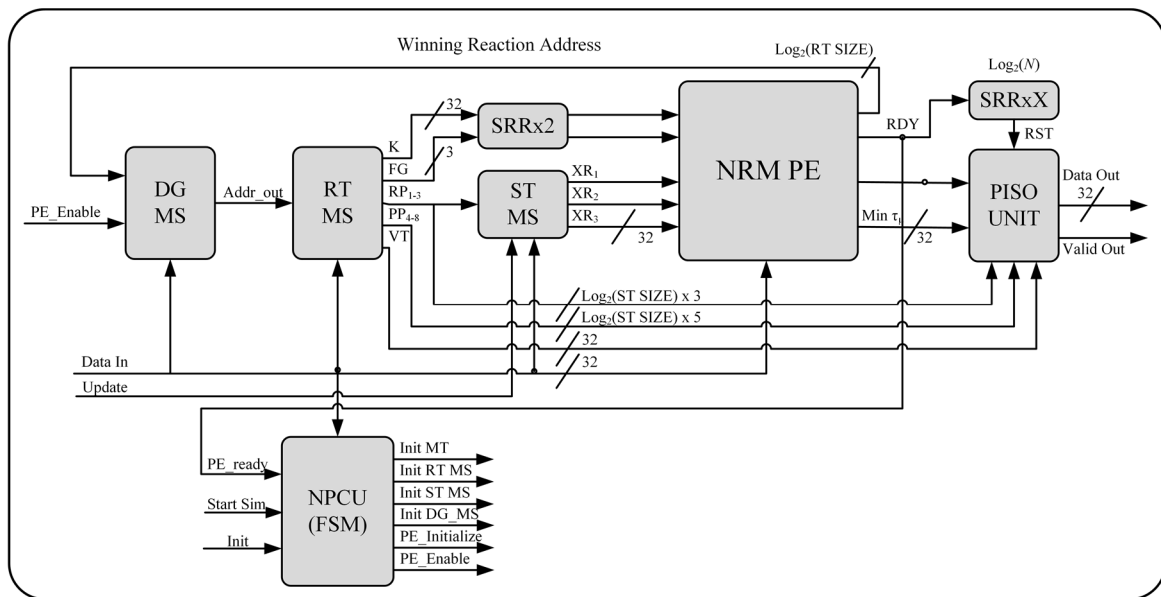


Εικόνα 6: Το RTL σχηματικό της μονάδας NPU.

Ο Πίνακας 2 που ακολουθεί εμπεριέχει όλα τα σήματα εισόδου και εξόδου της μονάδας NPU.

Πίνακας 2: Τα σήματα εισόδου και εξόδου της μονάδας NPU

Όνομα Σήματος	Περιγραφή
<b>Είσοδος</b>	
init	Όταν είναι ενεργοποιημένο πραγματοποιείται διαδικασία αρχικοποίησης και τα δεδομένα από την είσοδο πρέπει να προωθηθούν στην μονάδα που ορίζει η μονάδας ελέγχου.
din	Το κανάλι εισόδου των δεδομένων από το υψηλότερο επίπεδο προς την NPU και αντίστοιχα προς τις εσωτερικές μονάδες της.
start_sim	Όταν είναι ενεργοποιημένο η προσομοίωση εκτελείται κανονικά.
update	Όταν ενεργοποιηθεί τότε στο κανάλι εισόδου έχουν έρθει οι ανανεωμένες τιμές των μοριακών ειδών. Οι νέες τιμές θα πρέπει να αποθηκευτούν στην εσωτερική δομή της NPU ώστε να είναι διαθέσιμες για τον επόμενο κύκλο αντίδρασης.
<b>Εξόδος</b>	
m_axis_tdata	Το κανάλι εξόδου δεδομένων από την NPU προς το υψηλότερο επίπεδο σχεδίασης.
m_axis_tvalid	Όταν ενεργοποιηθεί ενημερώνει το υψηλότερο επίπεδο σχεδίασης ότι τα δεδομένα στο κανάλι εξόδου της NPU έχουν χρήσιμες πληροφορίες.



Εικόνα 7: Η αρχιτεκτονική της μονάδας NRM Processing Unit (NPU).

Όπως και στο επίπεδο πυρήνα, έτσι και στο επίπεδο επεξεργαστικής μονάδας, επαναχρησιμοποιούνται όλες οι βασικές μονάδες, προκειμένου να ελαχιστοποιηθεί ο απαιτούμενος χρόνος σχεδίασης. Έτσι λοιπόν, στο επίπεδο σχεδίασης της NPU (NRM Processing Unit) εμπεριέχονται όλες οι μονάδες που εμπεριέχονται στην μονάδα FPU (FRM Processing Unit), ελαφρώς τροποποιημένες. Τέλος, στην NPU έχει σχεδιαστεί και προστεθεί μια επιπλέον μονάδα διαχείρισης του γράφου εξαρτήσεων (Dependency Graph Memory System – DG MS).

Στην Εικόνα 7 παρουσιάζεται η αρχιτεκτονική της μονάδας NPU, ενώ η λειτουργία της και η ροή δεδομένων παρουσιάζεται εκτενώς στην ενότητα 5.5.

Στο επίπεδο της NPU, οι μονάδες που παραμένουν ίδιες και έχουν αναλυθεί εκτενώς στο Παραδοτέο 1.1 είναι οι εξής:

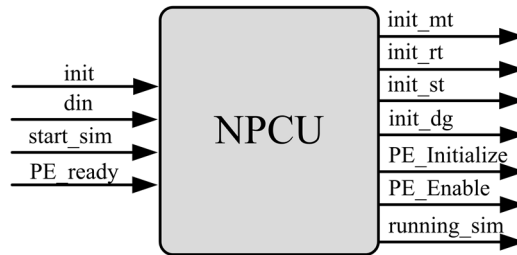
- Reaction Table Memory System (RT\_MS): Η μονάδα που διατηρεί σε μια δομή δεδομένων όλες τις πληροφορίες για τις αντιδράσεις του βιομοντέλου.
- ST Memory System (ST\_MS): Η μονάδα που διατηρεί σε μια δομή δεδομένων τους πληθυσμούς των μοριακών ειδών του βιομοντέλου καθ' όλη τη διάρκεια μιας επανάληψης της προσομοίωσης.
- Parallel in Serial out (PISO): Μια βασική μονάδα που δέχεται παράλληλα δεδομένα στις εισόδους της τα οποία στη συνέχεια οδηγεί σειριακά στην έξοδο της, και συγκεκριμένα στην έξοδο της NPU.

Ενώ, οι μονάδες που προστίθενται ή τροποποιούνται είναι οι εξής:

- NRM Processing Control Unit (NPCU): Τροποποιείται ελαφρώς. Προστίθεται μια επιπλέον κατάσταση στην FSM, ώστε να εκτελείται κατάλληλα η αρχικοποίηση του γράφου εξαρτήσεων.
- NRM Processing Element (NPE): Πλήρης επανασχεδιασμός. Προστίθεται ένα δέντρο συγκριτών και τροποποιείται η ροή δεδομένων μέσα στην υπολογιστική μονάδα του πυρήνα.
- Dependency Graph Memory System (DG\_MS): Η μονάδα που χειρίζεται τη δομή δεδομένων του γράφου εξαρτήσεων. Υπεύθυνη να εξάγει τις διευθύνσεις της νικητήριας και των επηρεαζόμενων αντιδράσεων.

Στη συνέχεια αναλύονται όλες οι μονάδες που προστέθηκαν ή τροποποιήθηκαν κατά τον σχεδιασμό της μονάδας NPU.

## 5.2 NRM Processing Control Unit – NPCU



Εικόνα 8: Το RTL σχηματικό της μονάδας NPCU.

Η μονάδα ελέγχου της υπολογιστικής οντότητας NPU, η NRM Processing Control Unit (NPCU), στο μεγαλύτερο μέρος της παραμένει ίδια με την FPCU (FRM Processing Control Unit) εκτός από τρία σημεία. Αρχικά, στις παραμέτρους του συστήματος προστίθενται μια ακόμα λέξη που δηλώνει το πλήθος των λέξεων που πρέπει να περιμένει το σύστημα ώστε να ολοκληρωθεί επιτυχώς η αρχικοποίηση του γράφου εξαρτήσεων. Εν συνεχεία, στην FSMD της μονάδας προστίθεται μια επιπλέον κατάσταση που σχετίζεται με την αρχικοποίηση του γράφου εξαρτήσεων. Στην κατάσταση αυτή η είσοδος του πυρήνα προωθείται στην είσοδο της μνήμης δεδομένων του γράφου και ενεργοποιείται το σήμα ελέγχου *init\_dg* που σηματοδοτεί την ενεργοποίηση εγγραφής στη μνήμη. Το σύστημα θα παραμείνει στην κατάσταση αυτή μέχρις ότου διαβαστεί όλο το πλήθος των απαιτούμενων λέξεων που έχει οριστεί από τις παραμέτρους του συστήματος. Τέλος, ο αλγόριθμος NRM, ως πρώτο βήμα, απαιτεί την αρχικοποίηση όλων των δομών δεδομένων που σχετίζονται με τις τάσεις  $\alpha_j$  και τους χρόνους  $\tau_j$  όλων των αντιδράσεων. Για να πραγματοποιηθεί η αρχικοποίηση, η υπολογιστική μονάδα (NPM Processing Element - NPE) του πυρήνα καλείται να υπολογίσει για μια και μόνο φορά τις τάσεις  $\alpha_j$  και τους χρόνους  $\tau_j$  όλων των αντιδράσεων και να αποθηκεύσει τις τιμές αυτές στις κατάλληλες δομές δεδομένων. Μόλις ολοκληρωθεί η διαδικασία αρχικοποίησης, έχει προκύψει η πρώτη νικητήρια αντίδραση και το σύστημα είναι πλέον σε θέση να λειτουργήσει. Στο εξής, το NPE υπολογίζει τους χρόνους  $\tau_j$  μόνο της νικητήριας αντίδρασης και των επηρεαζόμενων από αυτή αντιδράσεων. Για το λόγο αυτό, λοιπόν, η NPCU έχει σχεδιαστεί να παράγει δύο σήματα ελέγχου, το *PE\_initialize*, όπου τα δεδομένα όλων των αντιδράσεων προωθούνται στο NPE προκειμένου να αρχικοποιηθούν οι δομές δεδομένων, και το *PE\_enable*, όπου το NPE δέχεται μόνο τα δεδομένα της νικητήριας αντίδρασης και των επηρεαζόμενων αντιδράσεων. Και στις δύο περιπτώσεις, στο τέλος, προκύπτει μια νικητήρια αντίδραση, η αντίδραση δηλαδή με τον μικρότερο χρόνο ενεργοποίησης.

Ο πίνακας που ακολουθεί εμπεριέχει όλα τα σήματα εισόδου και εξόδου της μονάδας NPCU.



Πίνακας 3: Τα σήματα εισόδου και εξόδου της μονάδας NPCU.

Όνομα Σήματος	Περιγραφή
<b>Είσοδος</b>	
init	Όταν είναι ενεργοποιημένο πραγματοποιείται διαδικασία αρχικοποίησης και τα δεδομένα από την είσοδο πρέπει να προωθηθούν στην μονάδα που ορίζει η μονάδας ελέγχου.
din	Το κανάλι εισόδου των δεδομένων από το υψηλότερο επίπεδο προς την NPU και αντίστοιχα προς τις εσωτερικές μονάδες της.
start_sim	Όταν είναι ενεργοποιημένο η προσομοίωση εκτελείται κανονικά.
PE_ready	Ενημερώνει την NPCU ότι ένας κύκλος αντίδρασης έχει ολοκληρωθεί και μια νέα νικητήρια αντίδραση έχει προκύψει. Επίσης, το σήμα αυτό ενεργοποιεί τον μηχανισμό όπου συλλέγονται τα δεδομένα της νικητήριας αντίδρασης και μέσω της μονάδας PISO προωθούνται στην έξοδο της NPU.
<b>Έξοδος</b>	
init_mt	Σηματοδοτεί την αρχικοποίηση της δομής δεδομένων που σχετίζεται με τη γεννήτρια τυχαίων αριθμών, Mersenne Twister. Η δομή αυτή θα αρχικοποιηθεί μια και μόνο φορά, πριν την εκτέλεση της προσομοίωσης του μοντέλου.
init_rt	Σηματοδοτεί την αρχικοποίηση της δομής δεδομένων που σχετίζεται με τις πληροφορίες των αντιδράσεων, RT_MS. Η δομή αυτή θα αρχικοποιηθεί μια και μόνο φορά, πριν την εκτέλεση της προσομοίωσης του μοντέλου.
init_st	Σηματοδοτεί την αρχικοποίηση της δομής δεδομένων που σχετίζεται με τους πληθυσμούς των στοιχείων του μοντέλου. Η δομή αυτή αρχικοποιείται στην αρχή κάθε επανάληψης της προσομοίωσης.
init_dg	Σηματοδοτεί την αρχικοποίηση της δομής δεδομένων που σχετίζεται με τον γράφο εξαρτήσεων, DG_MS. Η δομή αυτή θα αρχικοποιηθεί μια και μόνο φορά, πριν την εκτέλεση της προσομοίωσης του μοντέλου.
PE_Initialize	Ενημερώνει τη μονάδα NPE ότι στα κανάλια εισόδου έχουν προωθηθεί τα δεδομένα (K, X, FG, κ.α.) όλων των αντιδράσεων. Ακολουθεί η αρχικοποίηση των εσωτερικών δομών δεδομένων του NPE.
PE_Enable	Ενημερώνει τη μονάδα NPE ότι στα κανάλια εισόδου έχουν προωθηθεί τα δεδομένα (K, X, FG, κ.α.) της νικητήριας αντίδρασης και των επηρεαζόμενων αντιδράσεων. Ακολουθούν όλες οι απαραίτητες πράξεις και συγκρίσεις ώστε να προκύψει η επόμενη νικητήρια αντίδραση.
running_sim	Ενεργοποιείται στην αρχή κάθε κύκλου αντιδράσεων ώστε να διαβαστούν όλα τα δεδομένα (K, X, FG, κ.α.) από τις κατάλληλες δομές δεδομένων και να προωθηθούν στα αντίστοιχα κανάλια εισόδου της μονάδας NPE.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ

ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

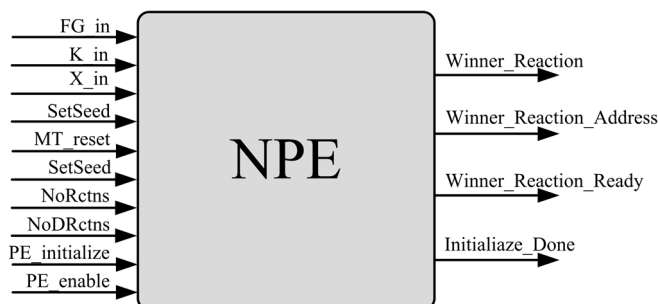
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
επένδυση στην κοινωνία της γνώσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

### 5.3 NRM Processing Element – NPE

Στο επίπεδο της NPU, το οποίο εξετάζουμε σε αυτήν την ενότητα, η οντότητα NPE αν και έχει δεχθεί ελάχιστες αλλαγές ως προς τα σήματα εισόδου/εξόδου, εσωτερικά, όμως, έχει επανασχεδιαστεί ολοκληρωτικά. Η αρχιτεκτονική της μονάδας NPE θα αναλυθεί στο Κεφάλαιο 6.



Εικόνα 9: Το RTL σχηματικό της μονάδας NPE.

Ο πίνακας που ακολουθεί εμπεριέχει όλα τα σήματα εισόδου και εξόδου της μονάδας NPE.

Πίνακας 4: Τα σήματα εισόδου και εξόδου της μονάδας NPE.

Όνομα Σήματος	Περιγραφή
<b>Είσοδος</b>	
FG_in	Παράμετρος που επηρεάζει τον υπολογισμό της τάσης $\alpha_j$ της εκάστοτε αντίδρασης.
K_in	Παράμετρος που επηρεάζει τον υπολογισμό της τάσης $\alpha_j$ της εκάστοτε αντίδρασης.
X_in	Ο πληθικός αριθμός των αντιδρώντων της εκάστοτε αντίδρασης.
MT_reset	Σήμα ελέγχου επαναφοράς της μονάδας παραγωγής τυχαίων αριθμών.
SetSeed	Σήμα ελέγχου εγγραφής στη δομή δεδομένων της μονάδας παραγωγής τυχαίων αριθμών.
Seed	Το κανάλι εισόδου δεδομένων της μονάδας παραγωγής τυχαίων αριθμών.
NoRctns	Το πλήθος των αντιδράσεων του μοντέλου που προσομοιώνεται.
NoDRctns	Το πλήθος των επηρεαζόμενων αντιδράσεων της νικητήριας αντίδρασης στον τρέχοντα κύκλο αντιδράσεων.
PE_initialize	Όταν ενεργοποιηθεί πραγματοποιείται η αρχικοποίηση των εσωτερικών δομών δεδομένων τη μονάδας NPE.
PE_enable	Το σήμα ενεργοποίησης της μονάδας NPE.
<b>Έξοδος</b>	
Winner_Reaction	Ο χρόνος της νικητήριας αντίδρασης.
Winner_Reaction_Address	Η διεύθυνση της νικητήριας αντίδρασης.
Winner_Reaction_Ready	Σηματοδοτεί την ολοκλήρωση της διαδικασίας εύρεσης νικητήριας αντίδρασης.
Initialize_Done	Σηματοδοτεί την ολοκλήρωση της αρχικοποίησης των δομών δεδομένων μέσα στη μονάδα NPE.

Η μονάδα NPE δέχεται στην είσοδό της όλες τις απαραίτητες πληροφορίες που απαιτούνται προκειμένου να υπολογιστεί η χρονική στιγμή που θα εκτελεστεί μια αντίδραση. Για να πραγματοποιηθούν οι κατάλληλες πράξεις υπολογισμού οι πληροφορίες της νικητήριας αντίδρασης και των επηρεαζόμενων αντιδράσεων εξάγονται από τη μνήμη του συστήματος και προωθούνται στην είσοδο της μονάδας NPE. Σημαντικό ρόλο στην εξαγωγή των δεδομένων από τη μνήμη παίζει ο γράφος εξαρτήσεων.

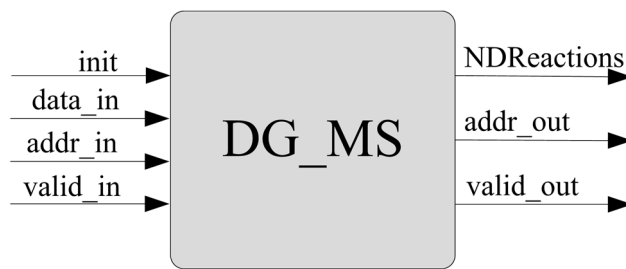
## 5.4 Dependency Graph (DG\_MS)

Ο γράφος εξαρτήσεων, Dependency Graph, είναι αποθηκευμένος σε μια δομή δεδομένων μέσα στη μονάδα Dependency Graph Memory System (DG\_MS). Η μονάδα DG\_MS, εκτός από τον γράφο, περιλαμβάνει μια απλή FSMD που είναι υπεύθυνη να δέχεται τη διεύθυνση της νικητήριας αντίδρασης και να κατευθύνει στην έξοδο τις διευθύνσεις των επηρεαζόμενων αντιδράσεων, τη μία μετά την άλλη. Οι διευθύνσεις αυτές προωθούνται στην μονάδα RT\_MS όπου εξάγονται οι πληροφορίες της αντίδρασης και οι διευθύνσεις των στοιχείων που συμμετέχουν στην εκάστοτε αντίδραση. Αφού διαβαστούν και οι πληθυσμοί των μοριακών ειδών, με τη βοήθεια της μονάδας ST\_MS, όλες οι πληροφορίες καταλήγουν στην είσοδο της μονάδας NPE.

Ακολουθεί ο πίνακας με τα σήματα εισόδου και εξόδου της μονάδας DG\_MS.

Πίνακας 5: Τα σήματα εισόδου και εξόδου της μονάδας DG\_MS.

Όνομα Σήματος	Περιγραφή
<b>Είσοδος</b>	
init	Όταν ενεργοποιηθεί πραγματοποιείται η αρχικοποίηση της μνήμης δεδομένων του γράφου. Τα δεδομένα από την είσοδο θα πρέπει να αποθηκευτούν στη μνήμη της μονάδας.
data_in	Το κανάλι εισόδου δεδομένων. Αφορούν μόνο τα δεδομένα αρχικοποίησης της μνήμης του γράφου.
addr_in	Η διεύθυνση της νικητήριας αντίδρασης.
valid_in	Όταν είναι ενεργοποιημένο η είσοδο addr_in έχει τη διεύθυνση της νικητήριας αντίδρασης και θα πρέπει η μονάδα DG_MS να εξάγει τις διευθύνσεις την νικητήριας αντίδρασης.
<b>Έξοδος</b>	
NDReactions	Στο κανάλι αυτό προωθούνται αρχικά η διεύθυνση της νικητήριας αντίδρασης και στη συνέχεια, η μια μετά την άλλη, οι διευθύνσεις των επηρεαζόμενων αντιδράσεων.
addr_out	Όταν το σήμα valid_out είναι ενεργοποιημένο τότε στην έξοδο της μονάδας έχουν εμφανιστεί διευθύνσεις που πρέπει να προωθηθούν τόσο στη μνήμη RT_MS όσο και στην μονάδα NPE.
valid_out	Μαζί με τις διευθύνσεις των αντιδράσεων η μονάδα DG_MS φροντίζει να ενημερώσει την μονάδα NPE για το πλήθος των επηρεαζόμενων αντιδράσεων της εκάστοτε νικητήριας αντίδρασης.



Εικόνα 10: Το RTL σχηματικό της μονάδας DG\_MS.

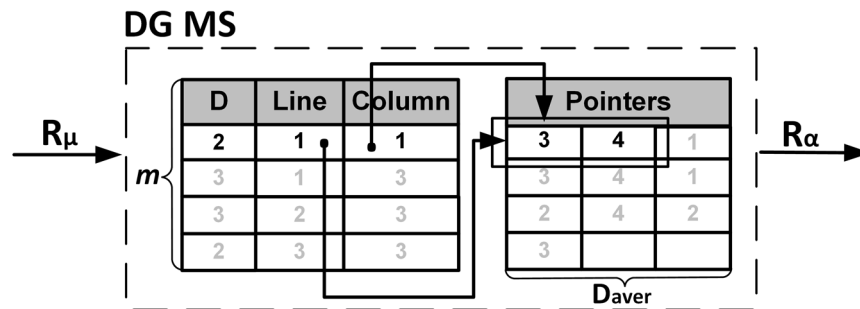
Η μονάδα DG\_MS αποτελείται από μια δομή δεδομένων που διατηρεί τα δεδομένα του γράφου σε μορφή πίνακα, μια μονάδα PISO (Parallel-In to Serial-Out) που συμμετέχει στην αρχικοποίηση του πίνακα και μια FSMD που με βάση την είσοδο είναι υπεύθυνη να προωθήσει στην έξοδο τις διευθύνσεις των κατάλληλων αντιδράσεων.

Η δομή δεδομένων του γράφου εξαρτήσεων έχει τη μορφή ενός πίνακα και ακολουθείται μια συγκεκριμένη στρατηγική ώστε να μεγιστοποιηθεί η αποτελεσματικότητα και να εκμεταλλευτούμε στο μέγιστο το μέγεθος της μνήμης που επιλέγουμε να έχουμε στη διάθεσή μας. Ο πίνακας του γράφου έχει βάθος (αριθμός γραμμών) ίσο με το πλήθος των αντιδράσεων ( $m$ ) του βιομοντέλου και επομένως η διεύθυνση της εκάστοτε αντίδρασης αντιστοιχεί στην ίδια διεύθυνση του πίνακα εξαρτήσεων. Δεδομένου ότι δεν έχουν όλες οι αντιδράσεις ίσο πλήθος επηρεαζόμενων αντιδράσεων, πρέπει αφενός να αποθηκευτεί το πλήθος των επηρεαζόμενων αντιδράσεων ( $D$ ) της κάθε αντίδρασης και αφετέρου να αποθηκευτούν όλες οι διευθύνσεις (*pointers*) των επηρεαζόμενων αντιδράσεων της εκάστοτε αντίδρασης. Οι διευθύνσεις αποθηκεύονται η μία μετά την άλλη χρησιμοποιώντας συνεχόμενες θέσεις μέσα στον πίνακα, ώστε να ελαχιστοποιηθεί η απαιτούμενη μνήμη. Επειδή, όμως, οι διευθύνσεις των επηρεαζόμενων αντιδράσεων δεν ξεκινούν από μια συγκεκριμένη θέση, χρησιμοποιούνται δύο στήλες του πίνακα ως ευρετήριο. Στο ευρετήριο αποθηκεύεται η γραμμή (*line*) και η στήλη (*column*) από όπου θα ξεκινήσει η FSMD να εξάγει τις απαιτούμενες διευθύνσεις.

Έτσι, λοιπόν, ο πίνακας του γράφου εξαρτήσεων αποτελείται από δύο τμήματα. Το πρώτο τμήμα αποτελείται από τρεις στήλες. Στην πρώτη στήλη αποθηκεύεται το πλήθος των επηρεαζόμενων αντιδράσεων της εκάστοτε αντίδρασης. Στη δεύτερη και στη τρίτη στήλη αποθηκεύεται το ευρετήριο για το τμήμα του γράφου. Το δεύτερο τμήμα του πίνακα χρησιμοποιείται εξ ολοκλήρου για την αποθήκευση διευθύνσεων επηρεαζόμενων αντιδράσεων. Επομένως, το πλάτος του πίνακα ορίζεται ως  $(D_{aver} + 3) * \log(m)$ , όπου 3 στήλες χρησιμοποιούνται για το ευρετήριο και  $D_{aver}$  στήλες για τις διευθύνσεις των επηρεαζόμενων αντιδράσεων. Η παράμετρος  $D_{aver}$  ορίζεται ως γενική παράμετρος (*generic*) του πυρήνα και θα πρέπει να καλύπτει το μέσο όρο των επηρεαζόμενων αντιδράσεων του μοντέλου. Κατά αυτό τον τρόπο ορίζεται η βέλτιστη τιμή ώστε να ελαχιστοποιείται η απαιτούμενη μνήμη πριν τη σύνθεση του πυρήνα.

Στην Εικόνα 11 παρουσιάζεται ένα ενδεικτικό παράδειγμα πίνακα ενός γράφου εξαρτήσεων για ένα μοντέλο με 4 αντιδράσεις, όπου οι εξαρτήσεις ανά αντίδραση είναι ως εξής:

- Η αντίδραση  $R1$  επηρεάζει τις αντιδράσεις  $R3$  και  $R4$ .
- Η αντίδραση  $R2$  επηρεάζει τις αντιδράσεις  $R1$ ,  $R3$ , και  $R4$ .
- Η αντίδραση  $R3$  επηρεάζει τις αντιδράσεις  $R1$ ,  $R2$ , και  $R4$ .
- Η αντίδραση  $R4$  επηρεάζει τις αντιδράσεις  $R2$  και  $R3$ .



Εικόνα 11: Ενδεικτικό παράδειγμα λειτουργίας γράφου εξαρτήσεων.

Στο παραπάνω παράδειγμα, ο μέσος όρος των επηρεαζόμενων αντιδράσεων είναι 2,5 αντιδράσεις. Για τον λόγο αυτό ο πίνακας έχει πλάτος 3 θέσεων ώστε να καλύψει τον μέσο όρο. Επίσης, είναι φανερό ότι για τις περισσότερες αντιδράσεις οι διευθύνσεις των επηρεαζόμενων αντιδράσεων επεκτείνονται σε περισσότερες από μια γραμμές του πίνακα. Προκειμένου η μονάδα DG\_MS να εξάγει τις διευθύνσεις, τη μια μετά την άλλη, φροντίζει πάντα να έχουν διαβαστεί δύο συνεχόμενες γραμμές του πίνακα. Αν για παράδειγμα, μια αντίδραση χρειάζεται τρεις γραμμές ώστε να αποθηκευτούν οι επηρεαζόμενες αντιδράσεις της, η μονάδα DG\_MS διαβάζει τις δύο πρώτες γραμμές και μόλις καλύψει όλες τις θέσεις της πρώτης γραμμής τότε διαβάζει την τρίτη γραμμή. Με αυτόν τον τρόπο η FSMD έχει στη διάθεσή της πάντα δύο συνεχόμενες γραμμές του πίνακα και μπορεί να εξάγει συνεχόμενες διευθύνσεις στην έξοδο της μονάδας DG\_MS.

Έτσι λοιπόν, εφόσον έχει πραγματοποιηθεί η αρχικοποίηση του συστήματος, η ροή των δεδομένων μέσα στη μονάδα που διαχειρίζεται τον γράφο εξαρτήσεων ακολουθεί τα εξής βήματα:

1. Στην είσοδο *addr\_in* έχει προωθηθεί η διεύθυνση της επιθυμητής νικητριάς αντίδρασης.
2. Στην αρχή ενός κύκλου αντιδράσεων ενεργοποιείται το σήμα *valid\_in* που σηματοδοτεί την έναρξη της εξαγωγής διευθύνσεων των επηρεαζόμενων αντιδράσεων της νικητριάς αντίδρασης από τον γράφο εξαρτήσεων.
3. Η FSMD της μονάδας DG\_MS διαβάζει την γραμμή του πίνακα που αντιστοιχεί στη νικητρία αντίδραση. Προωθεί στην έξοδο της μονάδας το πλήθος των επηρεαζόμενων αντιδράσεων και με βάση τη τιμή «line» του ευρετηρίου διαβάζει τις γραμμές του πίνακα με διεύθυνση «line» και «line+1», ώστε να έχει στη διάθεσή της δύο συνεχόμενες γραμμές του πίνακα.
4. Αναλόγως του πλήθους των επηρεαζόμενων αντιδράσεων η FSMD αρχίζει να προωθεί (*valid\_out*) στην έξοδο της μονάδας DG\_MS τις διευθύνσεις τους (*addr\_out*), τη μία μετά την άλλη. Είναι απαραίτητο να εμφανιστεί στην έξοδο πρώτα η διεύθυνση της νικητριάς αντίδρασης και στη συνέχεια οι διευθύνσεις των επηρεαζόμενων αντιδράσεων.
5. Η μονάδα DG\_MS θα σταματήσει να προωθεί διευθύνσεις επηρεαζόμενων αντιδράσεων στην έξοδο της μονάδας μόλις διαβαστούν από τον πίνακα τόσες διευθύνσεις όσο και το πλήθος των επηρεαζόμενων αντιδράσεων της αντίστοιχης νικητριάς αντίδρασης.
6. Η μονάδα DG\_MS απενεργοποιείται έως ότου ξεκινήσει ένα νέος κύκλος αντιδράσεων.

Η μονάδα DG\_MS είναι μια από τις σημαντικότερες μονάδες ενός πυρήνα NRM καθώς η έννοια της εξάρτησης των αντιδράσεων είναι εκείνη που κάνει τον αλγόριθμο NRM των Gibson και Bruke να διαφέρει αισθητά από τον αλγόριθμο FRM του Gillespie. Είναι εξίσου σημαντική, όμως, και ως προς τη σχεδίαση του πυρήνα καθώς καταλαμβάνει ένα μεγάλο ποσοστό των μνημών που έχουμε στη διάθεσή μας. Είναι σημαντικό, λοιπόν, να



δίδεται ιδιαίτερη σημασία στο μέσο όρο των επηρεαζόμενων αντιδράσεων και στον ορισμό της γενικής παραμέτρου DG\_SIZE πριν τη σύνθεση του πυρήνα.

## 5.5 Ροή Δεδομένων – NPU Datapath

Έχοντας αναλύσει τη λειτουργικότητα όλων των μονάδων, ολοκληρώνεται η περιγραφή της αρχιτεκτονικής της μονάδας NRM Processing Unit, NPU. Ακολουθεί η περιγραφή της ροής δεδομένων μέσα στην NPU κατά την αρχικοποίηση και κατά την προσομοίωση ενός μοντέλου.

Πριν από την εκκίνηση της προσομοίωσης ενός βιομοντέλου, το σύστημα θα πρέπει να αρχικοποιηθεί. Η διαδικασία αυτή περιλαμβάνει την αρχικοποίηση των παραμέτρων του συστήματος, όλων των δομών δεδομένων και όλων των σημάτων ελέγχου που σχετίζονται με την προσομοίωση. Η μονάδα ελέγχου NPCU είναι υπεύθυνη ώστε να προωθήσει κατάλληλα την πληροφορία από την είσοδο της NPU στην είσοδο των επιμέρους μονάδων, ενεργοποιώντας το σήμα *init* που αντιστοιχεί στην εκάστοτε μονάδα. Όταν ολοκληρωθεί η αρχικοποίηση των βασικών δομών δεδομένων σε αυτό το επίπεδο σχεδίασης, η NPCU ενεργοποιεί την αρχικοποίηση των δομών μέσα στην υπολογιστική μονάδα NPE. Οι δομές μέσα στη μονάδα NPE είναι ο πίνακας **T**, που διατηρεί την χρονική στιγμή ενεργοποίησης κάθε αντίδρασης (καθολικός χρόνος), και ο πίνακας **A**, που διατηρεί τις τάσεις  $\alpha_j$  για κάθε μια αντίδραση. Οι δύο αυτές δομές, όπως θα δούμε στην επόμενη ενότητα, είναι χρήσιμες για την ανανέωση του χρόνου  $\tau_j$  μιας επηρεαζόμενης αντίδρασης.

### Αρχικοποίηση:

Αρχικά, το σύστημα είναι σε κατάσταση αναμονής, περιμένοντας να ενεργοποιηθεί το σήμα *init* στην είσοδο της μονάδας NPU ώστε να ξεκινήσει η αρχικοποίηση. Όταν ενεργοποιηθεί, η μονάδα NPCU μεταβαίνει στην κατάσταση *set\_parameters* όπου ξεκινάει η αποθήκευση των παραμέτρων της προσομοίωσης. Οι απαραίτητοι παράμετροι προωθούνται από την είσοδο της NPU στην είσοδο της NPCU και είναι οι εξής:

- **Rt\_words**: το πλήθος των λέξεων (32 bit) που πρέπει η NPCU να διαβάσει και να προωθήσει στη δομή δεδομένων που διατηρεί τις πληροφορίες των αντιδράσεων του βιομοντέλου.
- **Number of Species**: το πλήθος των στοιχείων του βιομοντέλου.
- **Number of Repetitions**: το πλήθος των επαναλήψεων της προσομοίωσης του βιομοντέλου.
- **Dg\_words**: το πλήθος των λέξεων (32 bit) που πρέπει η NPCU να διαβάσει και να προωθήσει στη δομή δεδομένων που διατηρεί τις πληροφορίες του γράφου εξαρτήσεων των αντιδράσεων του μοντέλου.

Όταν όλες οι παράμετροι αποθηκευτούν τότε η NPCU μεταβαίνει στην κατάσταση *init\_mt\_state* και ξεκινάει η αρχικοποίηση της μονάδας παραγωγής τυχαίων αριθμών (Mersenne Twister Random Number Generator – MT RNG), η οποία χρησιμοποιεί τον αλγόριθμο Mersenne Twister [39]. Κατά το στάδιο αυτό η NPCU προωθεί την είσοδο της NPU στην είσοδο Seed της μονάδας NPE και παράγει το σήμα *init\_mt* που συνδέεται με το σήμα *SetSeed* της μονάδας NPE. Κατ' αυτόν τον τρόπο ενεργοποιείται η εγγραφή δεδομένων στους πίνακες της μονάδας MT. Για την μονάδα MT απαιτούνται 624 λέξεις ώστε να γεμίσουν οι πίνακες και να ολοκληρωθεί η διαδικασία. Στη συνέχεια, η NPCU μεταβαίνει στην κατάσταση *init\_rt\_state* και ξεκινάει η αρχικοποίηση της δομής που εμπεριέχει όλες τις πληροφορίες των αντιδράσεων. Η παράμετρος **Rt\_words** καθορίζει το πλήθος των λέξεων που περιμένει η NPCU να διαβάσει και να προωθήσει από την είσοδο της NPU στην είσοδο της μονάδας RT\_MS, ενεργοποιώντας αντίστοιχα τον σήμα *init\_rt* που στοχεύει στην εκκίνηση της εγγραφής των δεδομένων στον πίνακα RT. Αφού ολοκληρωθεί η αρχικοποίηση του πίνακα RT, το σύστημα προχωράει στην

αρχικοποίηση του πίνακα STI στο επίπεδο του πυρήνα, ένας πίνακας που στόχο έχει να βοηθάει στην αρχικοποίηση του πίνακα ST της NPU στην αρχή κάθε επανάλληψης (περισσότερες πληροφορίες στο Παραδοτέο 1.1). Η επόμενη κατάσταση στην οποία θα μεταβεί η NPCU είναι η *init\_dg\_state*, όπου με βάση το πλήθος των λέξεων που καθορίζει η παράμετρος *Dg\_words*, η NPCU θα στέλνει τα δεδομένα από την είσοδο της NPU στην είσοδο της μονάδας DG\_MS και θα ενεργοποιήσει το σήμα *init\_dg* που στοχεύει στην έναρξη της εγγραφής των δεδομένων στον πίνακα DG. Μετά την ολοκλήρωση της αρχικοποίησης της δομής του γράφου εξαρτήσεων, η NPCU μεταβαίνει σε μια κατάσταση αναμονής, μέχρις ότου το σύστημα να είναι έτοιμο να εκτελέσει την πρώτη επανάλληψη της προσομοίωσης.

#### Αρχικοποίηση της μονάδας NPE:

Αφού ολοκληρωθεί η αρχικοποίηση των βασικών δομών, το σύστημα είναι έτοιμο να εκτελέσει την πρώτη επανάλληψη της προσομοίωσης. Στην αρχή κάθε επανάλληψης θα πρέπει να πραγματοποιηθούν δύο διαδικασίες αρχικοποίησης. Η πρώτη διαδικασία αφορά τον πίνακα ST όπου αποθηκεύονται οι πληθυσμοί των μοριακών ειδών του μοντέλου καθ' όλη τη διάρκεια της προσομοίωσης. Η μονάδα SCU είναι υπεύθυνη να διαβάσει τις αρχικές τιμές από την μονάδα STI\_MS και να τις προωθήσει στην μονάδα NPU ώστε να αρχικοποιηθεί ο πίνακας ST. Η διαδικασία αυτή πραγματοποιείται στην αρχή κάθε επανάλληψης της προσομοίωσης.

Η δεύτερη διαδικασία αφορά την αρχικοποίηση των εσωτερικών δομών του NPE. Σύμφωνα με τον αλγόριθμο των Gibson και Bruke, για την ανανέωση του χρόνου  $\tau_j$  μια επηρεαζόμενη αντίδραση απαιτείται η προηγούμενη τιμή του χρόνου  $\tau_j$  και της τάσης  $\alpha_j$  της αντίδρασης αυτής. Για τον λόγο αυτό, μέσα στη μονάδα NPE διατηρούνται οι χρόνοι  $\tau_j$  και οι τάσεις  $\alpha_j$  όλων των αντιδράσεων, ώστε να χρησιμοποιηθούν αργότερα στις πράξεις ανανέωσης των χρόνων  $\tau_j$  των επηρεαζόμενων αντιδράσεων. Στην αρχή κάθε επανάλληψης, οι δομές αυτές είναι κενές, επομένως απαιτείται τα δεδομένα όλων των αντιδράσεων να προωθηθούν στην μονάδα NPE ώστε να αρχικοποιηθούν. Η μονάδα ελέγχου NPCU στην αρχή κάθε επανάλληψης ενεργοποιεί το σήμα *PE\_initialize* που σηματοδοτεί την εκκίνηση της διαδικασίας αυτής. Η μονάδα NPE, όταν ολοκληρωθεί η αρχικοποίηση των δομών δεδομένων, έχει αποφασίσει για την πρώτη νικητήρια αντίδραση και ενημερώνει την μονάδα NPCU για την ολοκλήρωση της αρχικοποίησης. Στο εξής, το NPE υπολογίζει τους χρόνους  $\tau_j$  μόνο της νικητήριας αντίδρασης και των επηρεαζόμενων αντιδράσεων. Περισσότερες πληροφορίες για το NPE θα δοθούν στο Κεφάλαιο 6 του παραδοτέου.

#### Κανονική λειτουργία:

Δεδομένου ότι έχει ολοκληρωθεί πλέον η αρχικοποίηση του συστήματος, ο πυρήνας NRM είναι σε θέση να εκτελέσει έναν ή περισσότερους κύκλους αντιδράσεων, μέχρις ότου ολοκληρώσει όλες τις επαναλήψεις για τον επιθυμητό χρόνο προσομοίωσης. Ένας κύκλος αντιδράσεων απαρτίζεται από μια συγκεκριμένη ροή δεδομένων μέσα στον πυρήνα και στις υπομονάδες του. Όπως αναφέρθηκε στην προηγούμενη παράγραφο, στο τέλος της αρχικοποίησης των εσωτερικών δομών της μονάδας NPE προκύπτει η πρώτη νικητήρια αντίδραση. Στην αρχή κάθε κύκλου αντιδράσεων η νικητήρια αντίδραση από τον προηγούμενο κύκλο αντιδράσεων είναι εκείνη που καθορίζει τα δεδομένα που θα προωθηθούν εκ νέου στην μονάδα NPE. Συγκεκριμένα, η μονάδα ελέγχου NPCU ενεργοποιεί την μονάδα DG\_MS η οποία δέχεται τη διεύθυνση της νικητήριας αντίδρασης και εξάγει από τον γράφο εξαρτήσεων τις διευθύνσεις όλων των επηρεαζόμενων αντιδράσεων. Στη συνέχεια οι διευθύνσεις και τα δεδομένα της νικητήριας αντίδρασης και των επηρεαζόμενων αντιδράσεων προωθούνται στην μονάδα NPE. Η μονάδα NPE, με βάση τον αλγόριθμο, κάνει τις απαραίτητες ενέργειες προκειμένου να ανανεωθούν οι χρόνοι

των παραπάνω αντιδράσεων και τελικά αποφασίζει ποια θα είναι η νέα νικητήρια αντίδραση από το σύνολο των αντιδράσεων του βιομοντέλου. Οι πληροφορίες για την νικητήρια αντίδραση προωθούνται στις μονάδες SPU και SCU, αφενός για να ανανεωθούν ο πληθυσμός των μοριακών ειδών που συμμετέχουν στην αντίδραση και αφετέρου για να υπολογιστεί ο τρέχον χρόνος προσομοίωσης. Ένας νέος κύκλος αντιδράσεων ξεκινάει και η ίδια ροή δεδομένων επαναλαμβάνεται όσες φορές χρειαστεί μέχρι η προσομοίωση να φτάσει στον επιθυμητό χρόνο προσομοίωσης ( $T_{sim}$ ) που έχει ορίσει ο χρήστης του συστήματος.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



## 6 Η ΚΕΝΤΡΙΚΗ ΜΟΝΑΔΑ ΕΠΕΞΕΡΓΑΣΙΑΣ - NRM PROCESSING ELEMENT

Η μονάδα NRM Processing Element (NPE) αποτελεί την κεντρική μονάδα επεξεργασίας και είναι μια από τις βασικότερες μονάδες του πυρήνα NRM. Όπως έχει αναφερθεί στην ενότητα 5.3, η μονάδα NPE δέχεται όλες τις πληροφορίες μιας αντίδρασης που απαιτούνται για να υπολογιστεί ο νέος χρόνος εκτέλεσης της αντίδρασης, είτε αυτή είναι η νικητήρια αντίδραση είτε είναι μια επηρεαζόμενη από τη νικητήρια αντίδραση.

### 6.1 Βασική Λειτουργία και Ροή Δεδομένων

Στη μονάδα NPE πραγματοποιούνται όλες οι αριθμητικές και λογικές πράξεις προκειμένου να προκύψει η νικητήρια αντίδραση κάθε κύκλου αντιδράσεων. Οι αριθμητικές πράξεις αφορούν την εφαρμογή των μαθηματικών τύπων του αλγορίθμου NRM, ενώ οι λογικές πράξεις είναι καθαρά πράξεις σύγκρισης και εύρεσης της αντίδρασης με τον μικρότερο αναμενόμενο χρόνο εκτέλεσης. Έτσι λοιπόν, η μονάδα NPE χωρίζεται σε δύο μονάδες, την Recalculation Unit (RU), που εκτελεί τις αριθμητικές πράξεις, και το Comparator Tree (CoT), που εκτελεί τις πράξεις σύγκρισης και εύρεσης της αντίδρασης με τον ελάχιστο χρόνο.

Τέλος, μέσα στη μονάδα NPE εμπεριέχονται δύο πολύ βασικές δομές δεδομένων:

- ο πίνακας **T** (Table T) στον οποίο αποθηκεύονται οι χρόνοι  $\tau_j$  όλων των αντιδράσεων.
- ο πίνακας **A** (Propensity Table) στον οποίο αποθηκεύονται οι τάσεις  $\alpha_j$  όλων των αντιδράσεων.

Ο πίνακας **T** είναι αναπόσπαστο κομμάτι της μονάδας CoT, ενώ ο πίνακας **A** εμπεριέχεται στην μονάδα RU, έτσι ώστε να διευκολύνεται η ανάγνωση και εγγραφή από και προς τους πίνακες αυτούς.

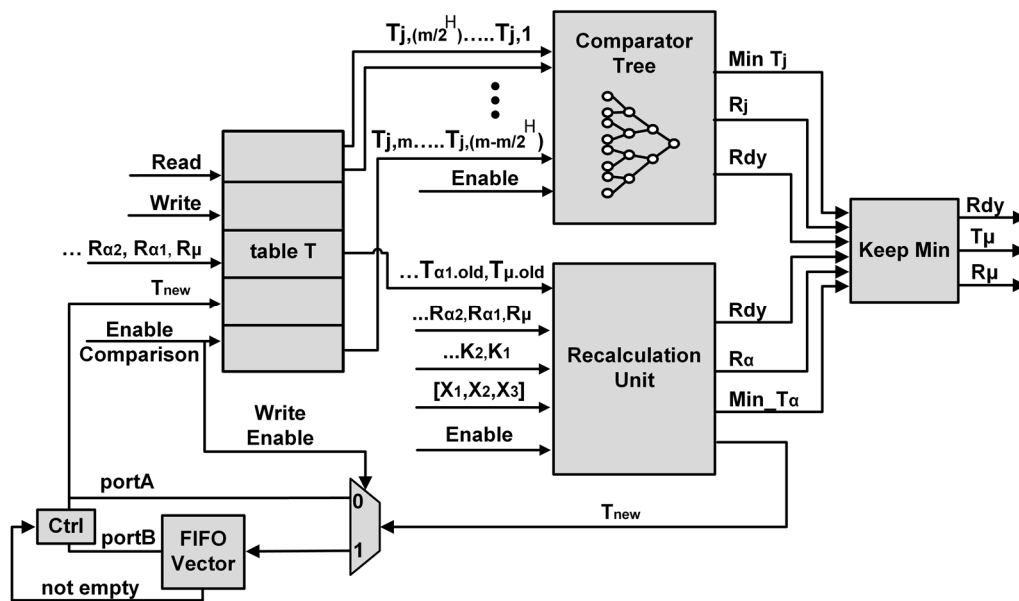
Σε κατάσταση αρχικοποίησης των δομών, η μονάδα RU ενεργοποιείται με το σήμα *PE\_initialize*, δέχεται στην είσοδό της όλες τις πληροφορίες για όλες τις αντιδράσεις, υπολογίζει και αποθηκεύει τις τάσεις  $\alpha_j$  όλων των αντιδράσεων, και τελικά υπολογίζει και προωθεί στον πίνακα **T** τους χρόνους  $\tau_j$  όλων των αντιδράσεων. Όσο η μονάδα RU παράγει χρόνους, τους συγκρίνει, έτσι ώστε στο τέλος της φάσης της αρχικοποίησης να έχει βρεθεί η πρώτη νικητήρια αντίδραση.

Κατά την εκτέλεση ενός κύκλου αντιδράσεων, ο πίνακας **T** τροφοδοτεί τη μονάδα RU με τους παλιούς χρόνους της νικητήριας αντίδρασης ( $\tau_{\mu,old}$ ) και των επηρεαζόμενων αντιδράσεων ( $\tau_{\alpha,old}$ ), ώστε οι χρόνοι αυτοί να ανανεωθούν και να αποθηκευτούν εκ νέου. Όσο η μονάδα RU κάνει τις απαιτούμενες πράξεις, η μονάδα CoT, χρησιμοποιώντας ένα δέντρο συγκριτών, συγκρίνει του χρόνους των υπολοίπων αντιδράσεων του πίνακα **T**. Η νέα νικητήρια αντίδραση προκύπτει από τη σύγκριση του αποτελέσματος της μονάδας CoT και από την μικρότερη τιμή ανανεωμένου χρόνου εκτέλεσης που έχει υπολογίσει η μονάδα RU.

Συνοπτικά τα βήματα που εκτελούνται για την εύρεση της αντίδρασης με τον μικρότερο χρόνο εκτέλεσης:

- Στην είσοδο της μονάδας NPE διοχετεύονται οι πληροφορίες για τη νικητήρια αντίδραση και τις επηρεαζόμενες από αυτήν αντιδράσεις.
- Με βάση τις διευθύνσεις των αντιδράσεων, οι χρόνοι  $\tau_{\mu,old}$  και  $\tau_{\alpha,old}$  εξάγονται από τον πίνακα **T**, προωθούνται στη μονάδα RU και τελικά στιγματίζονται ώστε να μην συμμετέχουν στις συγκρίσεις της μονάδας CoT.

- Στην είσοδο της μονάδας RU διοχετεύονται οι πληροφορίες για τη νικητήρια αντίδραση και τις επηρεαζόμενες από αυτήν αντιδράσεις. Καθώς τα δεδομένα εισέρχονται, η μονάδα RU αρχίζει να υπολογίζει τους καινούργιους χρόνους.
- Όταν προωθηθούν όλοι οι χρόνοι στην είσοδο της μονάδας RU, ενεργοποιείται η μονάδα CoT.
- Στο τέλος των μονάδων υπάρχει μια έξτρα λογική σύγκρισης που κρατάει την τρέχουσα ελάχιστη τιμή. Οι ελάχιστοι χρόνοι από τις δύο μονάδες συγκρίνονται και τελικά προκύπτει η νέα νικητήρια αντίδραση.
- Η διεύθυνση της νικητήριας αντίδρασης οδηγείται στην έξοδο και η μονάδα NPE μεταβαίνει σε κατάσταση αναμονής, έως ότου ξεκινήσει ο επόμενος κύκλος αντιδράσεων.



Εικόνα 12: Η αρχιτεκτονική της μονάδας NRM Process Element (NPE)

Τα παραπάνω βήματα υλοποιούνται από τη μονάδα NRM Processing Element (NPE) της οποίας η αρχιτεκτονική απεικονίζεται στην Εικόνα 12. Στη συνέχεια κεφαλαίου, γίνεται εκτενής περιγραφή των μονάδων RU και CoT, ενώ αναφέρονται περισσότερες λεπτομέρειες για τον τρόπο λειτουργίας της μονάδας NPE.

## 6.2 Πίνακας Χρόνων Ενεργοποίησης Αντιδράσεων

Στον πίνακα **T** αποθηκεύεται για κάθε αντίδραση η αναμενόμενη χρονική στιγμή ενεργοποίησής της. Σε κάθε κύκλο αντιδράσεων η αντίδραση με τον δυνητικά μικρότερο χρόνο είναι η «νικητήρια αντίδραση» ( $R_\mu$ ) και είναι αυτή που τελικά θα εκτελεστεί.

Ο πίνακας **T** και η μονάδα CoT συνθέτουν μια οντότητα που σκοπός της είναι η διαχείριση των χρόνων  $\{\tau_j\}$  όλων των αντιδράσεων. Η οντότητα αυτή υποστηρίζει τρεις λειτουργίες:

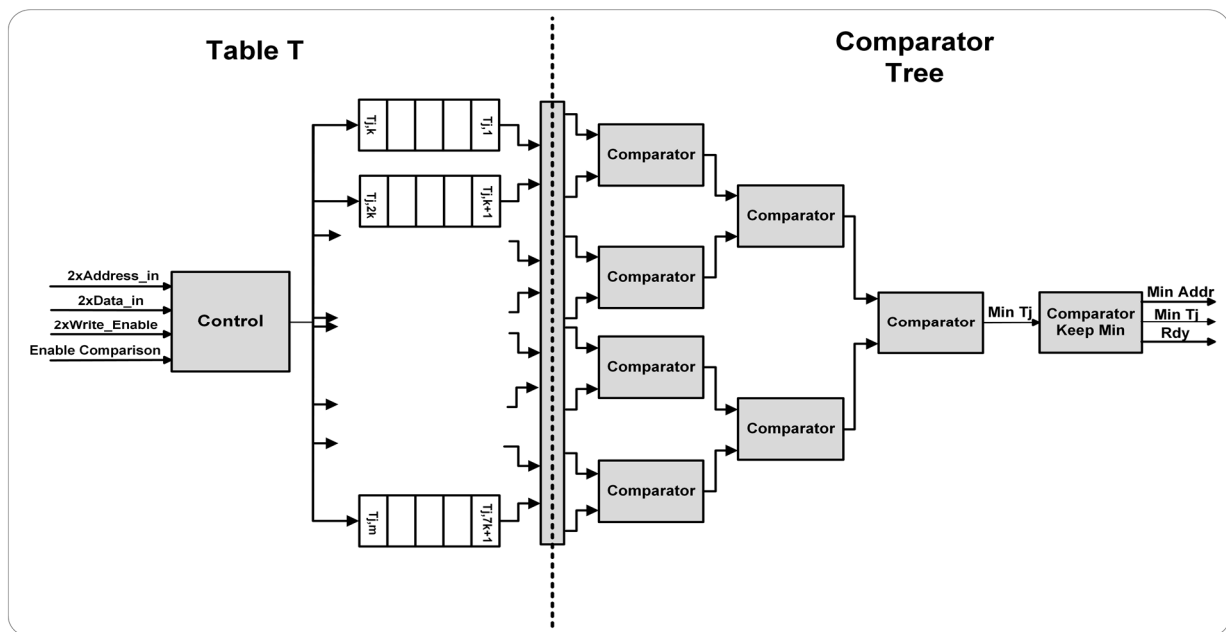
- Ανάγνωση\_Δεδομένων: Η ανάγνωση των χρόνων  $\{\tau_j\}$  των αντιδράσεων και η διοχέτευσή τους προς τη μονάδα RU.
- Εγγραφή\_Δεδομένων: Η εγγραφή των νέων χρόνων  $\{\tau_j\}$  που προκύπτουν από υπολογισμούς της

μονάδας RU. Ενεργοποιείται το σήμα εγγραφής δεδομένων *write\_enable* και χρησιμοποιείται είτε η μία είτε και οι δύο διαθέσιμες πόρτες εγγραφής δεδομένων.

- Λειτουργία Σύγκρισης: Ενεργοποιείται η μονάδα CoT (δέντρο συγκριτών) ώστε να βρεθεί ο ελάχιστος χρόνος  $\tau_j$  του πίνακα **T**. Ενεργοποιείται το σήμα *enable\_comparison* και τα δεδομένα του πίνακα διοχετεύονται σταδιακά στο δέντρο συγκριτών.

Δεδομένου ότι από μια μνήμη RAM μπορούμε να διαβάσουμε μέχρι δύο δεδομένα σε κάθε κύκλο ρολογιού, και προκειμένου να μεγιστοποιηθεί το επίπεδο της παραλληλίας, ο πίνακας **T** έχει καταταξιωθεί σε επιμέρους τμήματα. Ο λόγος είναι κυρίως για να πραγματοποιείται ταυτόχρονη διοχέτευση δεδομένων από τον πίνακα **T** στα φύλλα του δέντρου συγκριτών. Συνεπώς, αν  $2^H$  είναι το πλήθος των φύλλων του δέντρου, αντίστοιχα σε τόσα κομμάτια χωρίζεται και ο πίνακας **T**. Κάθε τμήμα του πίνακα αποθηκεύει  $m/2^H$  χρόνους αντιδράσεων και εν τέλει  $m/2^{(H+1)}$  κύκλοι ρολογιού χρειάζονται για να διοχετευτούν όλα τα δεδομένα του πίνακα **T** στα φύλλα του δέντρου συγκριτών.

Το σχηματικό της Εικόνας 13 απεικονίζει την μορφή του πίνακα **T** στην περίπτωση που το δέντρο έχει ύψος  $H = 3$ . Ο πίνακας **T** και το δέντρο συγκριτών ενώνονται με ένα κατάλληλου μεγέθους κανάλι δεδομένων που στόχο έχει να προωθήσει παράλληλα τα δεδομένα από τον πίνακα **T** στο δέντρο συγκριτών.



Εικόνα 13: Ενδεικτικό παράδειγμα του πίνακα **T** και του δέντρου συγκριτών (CoT) ύψους  $H = 3$ .

Η δομή δεδομένων του πίνακα **T**, για λόγους ασφαλείας, μπορεί να διαβάσει και να γράψει μέχρι δύο δεδομένα σε κάθε κύκλο ρολογιού ( $2xData\_in$ ,  $2xWrite\_Enable$ ,  $2xAddress\_in$ ), ενώ κατά τη διάρκεια συγκρίσεων (*enable\_comparison*) δεν επιτρέπονται η ανάγνωση και η εγγραφή δεδομένων από/προς τον πίνακα **T**. Με αυτόν τον τρόπο διασφαλίζεται η ορθή λειτουργία της μνήμης RAM με δύο πόρτες (Dual Port RAM – DPRAM) και αποφεύγεται η σύγκρουση διευθύνσεων.

### 6.3 Δέντρο Συγκριτών – Comparator Tree (CoT)

Η μονάδα CoT (Comparator Tree) είναι ένα δυαδικό δέντρο συγκριτών όπου κάθε φύλλο/κόμβος είναι ένας συγκριτής δύο εισόδων. Ρυθμίζοντας κατάλληλα το ύψος του δέντρου ρυθμίζουμε τον συνολικό αριθμό των κύκλων ρολογιού που απαιτούνται προκειμένου να προωθηθούν όλα τα δεδομένα από τον πίνακα **T** προς το δέντρο και να καθοριστεί η αντίδραση με το μικρότερο  $t_j$  του πίνακα **T**.

Όπως αναφέρθηκε προηγουμένως, ο πίνακας **T** και το δέντρο συγκριτών στην ουσία συνθέτουν μια οντότητα ώστε να απλοποιηθεί η σχεδίαση και να απλοποιηθεί η ροή των δεδομένων από τον πίνακα **T** προς τα φύλλα του δέντρου.

Τα μεγέθη που επηρεάζονται από την παράμετρο  $H$  είναι τα εξής:

- το πλήθος των φύλλων – συγκριτές ( $2^H$ )
- το πλήθος των τμημάτων που θα επιμεριστεί ο πίνακας **T** ( $2^H$ )
- το βάθος του κάθε τμήματος του πίνακα **T** ( $m/2^H$ )

Συνεπώς, το ύψος του δέντρου καθορίζει το πλήθος των κύκλων που απαιτούνται για να βρεθεί στον πίνακα **T** η αντίδραση με τον δυνητικά μικρότερο χρόνο ενεργοποίησης. Δεδομένου ότι  $m$  είναι το πλήθος των αντιδράσεων και  $H$  το ύψος του δέντρου, ορίζουμε ως  $L_{CoT}$  (CoT Latency) τη συνολική καθυστέρηση σε κύκλους ρολογιού (cc) που απαιτούνται για να ολοκληρωθεί η διαδικασία σύγκρισης και ισχύει ότι:

$$L_{CoT} = \frac{m}{2^{H+1}} + H + 1 \quad (5)$$

Οι όροι της εξίσωσης είναι οι εξής:

- το βάθος της μνήμης σε κάθε φύλλο ( $\frac{m}{2^{H+1}}$ )
- το ύψος του δέντρου ( $H$ )
- ένας κύκλος καθυστέρησης ανάγνωσης από την μνήμη (Πίνακα **T**) (+ 1 cc)

Τέλος, είναι σημαντικό να αναφερθεί ότι το ύψος του δέντρου ( $H$ ) ορίζεται πριν τη σύνθεση του πυρήνα και επιλέγεται κατάλληλα ώστε η τιμή  $L_{CoT}$  να μη ξεπερνά την συνολική καθυστέρηση της μονάδας RU, που όπως θα δούμε στη συνέχεια, ορίζεται ως  $L_{RU} = 42 + D$  cc. Κατά αυτόν τον τρόπο η μονάδα CoT λειτουργεί παράλληλα με τη μονάδα RU, επιτυγχάνοντας έτσι την απόκρυψη της επιπλέον καθυστέρησης που χρειάζεται ώστε να βρεθεί η αντίδραση του πίνακα **T** με τον μικρότερο χρόνο.

Ο πίνακας που ακολουθεί παρουσιάζει τη συνολική καθυστέρηση ( $L_{CoT}$ ) που απαιτείται για διαφορετικό πλήθος αντιδράσεων ( $m$ ) και διαφορετικό ύψος δέντρου ( $H$ ). Τα κελιά που το περιεχόμενό τους είναι ελαφρώς χρωματισμένα με γκριζό χρώμα αποτελούν ακραίες ή μη αποδεκτές τιμές για τη δική μας υλοποίηση. Βασική προϋπόθεση να ισχύει η ανισότητα  $L_{CoT} < L_{RU}$ .

Πίνακας 6: Το πλήθος των απαιτούμενων κύκλων ρολογιού ( $L_{CoT}$ ) συναρτήσει του πλήθους των αντιδράσεων  $m$  και του ύψους του δέντρου  $H$ .

$H$	1	2	3	4	5	6	7
$m$							
1	2	3	4	5	6	7	8
2	3	3	4	5	6	7	8
4	3	4	4	5	6	7	8
8	4	4	5	5	6	7	8
16	6	5	5	6	6	7	8
32	10	7	6	6	7	7	8
64	18	11	8	7	7	8	8
128	34	19	12	9	8	8	9
256	66	35	20	13	10	9	9
512	130	67	36	21	14	11	10
1024	258	131	68	37	22	15	12
2048	514	259	132	69	38	23	16
4096	1026	515	260	133	70	39	24

#### 6.4 Μονάδα Επανυπολογισμού – Recalculation Unit (RU)

Η μονάδα Recalculation Unit (RU) είναι κατάλληλα σχεδιασμένη ώστε να εκτελεί τους μαθηματικούς τύπους (3) και (4) του αλγορίθμου Next Reaction Method (NRM). Η αρχιτεκτονική της είναι παρόμοια με εκείνη της υπολογιστικής μονάδας FPU του FRM-SSA SoC που αναλύεται στο Παραδοτέο 1.1, με πολλές όμως αλλαγές όσο αναφορά τη ροή δεδομένων. Στη συνέχεια, αναλύεται διεξοδικά η ροή των δεδομένων και όλες οι σχεδιαστικές επιλογές, ώστε η μονάδα RU να πληροί τις προϋποθέσεις του αλγορίθμου NRM.

Θεωρώντας ότι:

- $t$  είναι ο χρόνος προσομοίωσης την τρέχουσα χρονική στιγμή,
- $\alpha_j$  η τάση (propensity) μιας αντίδρασης με βάση τα δεδομένα της αντίδρασης ( $K, FG$ ) και τον πληθυσμό των αντιδρώντων ( $X_1, X_2, X_3$ )
- $r_j$  ένας τυχαίος αριθμός στο διάστημα  $[0, 1]$ ,
- $\alpha_{j,old}$  η τάση (propensity) μιας αντίδρασης στον προηγούμενο κύκλο αντιδράσεων,
- $\tau_{j,old}$  ο χρόνος μιας αντίδρασης στον προηγούμενο κύκλο αντιδράσεων.

Ο μαθηματικός τύπος (3) χρησιμοποιείται για την ανανέωση του χρόνου της νικητήριας αντίδρασης και δεν διαφέρει από αυτόν που χρησιμοποιεί ο FRM αλγόριθμος.

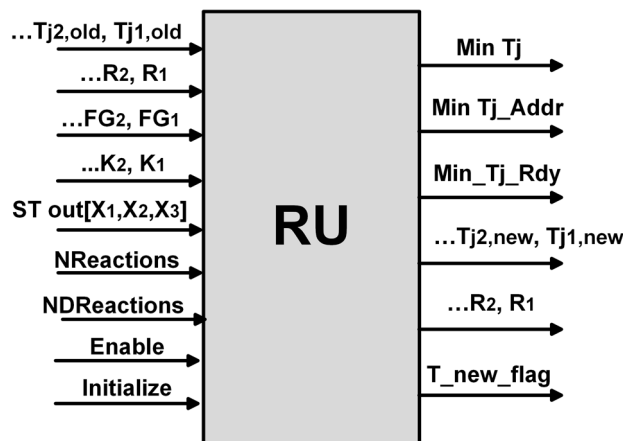
$$\tau_j = t + \frac{1}{\alpha_j(x)} \ln\left(\frac{1}{r_j}\right) \quad (3)$$

Ο μαθηματικός τύπος (4) χρησιμοποιείται για την ανανέωση του χρόνου μιας επηρεαζόμενης αντίδρασης.

$$\tau_j = t + \frac{\alpha_{j,old}}{\alpha_j} * (\tau_{j,old} - t) \quad (4)$$

Σε αυτό το σημείο χρειάζεται να αναφέρουμε τα εξής:

- Όλοι οι αριθμοί που συμμετέχουν στις πράξεις είναι αριθμοί κινητής υποδιαστολής απλής ακρίβειας (IEEE 754) [40].
- Για την παραγωγή ψευδο-τυχαίων αριθμών χρησιμοποιήθηκε μια μηχανή βασισμένη στον αλγόριθμο Mersenne Twister [39]. Για να παράγουμε τυχαίους αριθμούς στο πεδίο τιμών [0,1] θεωρούμε τον αριθμό που παράγεται ως σταθερής υποδιαστολής και τον μετατρέπουμε σε κινητής υποδιαστολής. Με την σωστή μετατροπή, ο τελικός τυχαίος αριθμός που προκύπτει είναι κινητής υποδιαστολής απλής ακρίβειας στο επιθυμητό διάστημα.
- Για τις απαραίτητες πράξεις αριθμών κινητής υποδιαστολής χρησιμοποιούνται διαθέσιμες μονάδες της εταιρίας Xilinx (Core Generator). Οι μονάδες αυτές υλοποιούν τον πολλαπλασιασμό, την αφαίρεση, την πρόσθεση, τον πολλαπλασιασμό, τη διαίρεση, τον φυσικό λογάριθμο και την μετατροπή αριθμών από σταθερή σε κινητή υποδιαστολή.
- Όσο αναφορά τον υπολογισμό της τάσης  $\alpha_j$  (propensity) μιας αντίδρασης του βιομοντέλου, επαναχρησιμοποιείται από το Παραδοτέο 1.1 η μονάδα Propensity Unit, η οποία δέχεται ως είσοδο τον πληθυσμό των αντιδρώντων και τις παραμέτρους FG και K μιας αντίδρασης και υπολογίζει την τάση της αντίδρασης (μέχρι  $3^{15}$  τάξης).
- Για την υλοποίηση του αλγορίθμου, οι τάσεις όλων των αντιδράσεων αποθηκεύονται σε έναν τοπικό πίνακα ώστε να επαναχρησιμοποιηθούν σε μετέπειτα υπολογισμούς. Ο πίνακας αυτός (Propensity Table) έχει βάθος  $m$ , όσες είναι οι αντιδράσεις του μοντέλου και υλοποιείται με μνήμες RAM (Dual Port).



Εικόνα 14: Το RTL σχηματικό της μονάδας Recalculation Unit (RU).



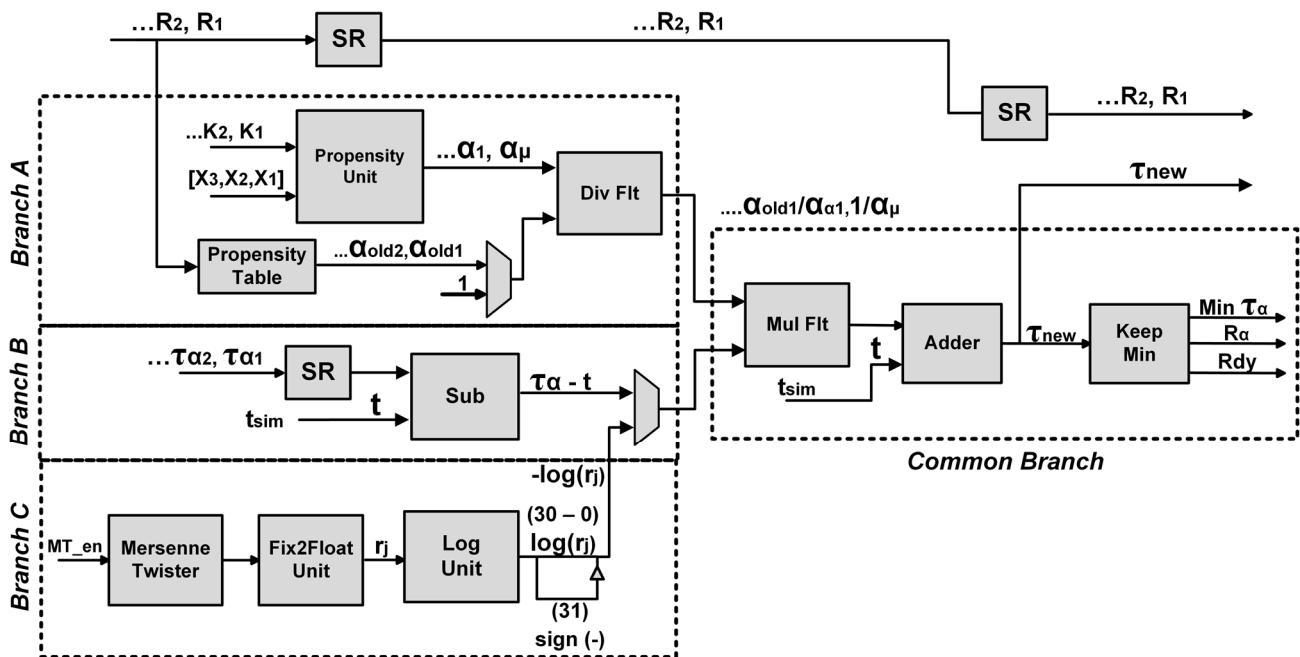
Πίνακας 7: Τα σήματα εισόδου και εξόδου της μονάδας Recalculation Unit (RU).

Όνομα Σήματος	Περιγραφή
<b>Είσοδος</b>	
Enable	Όταν είναι ενεργοποιημένο, η μονάδα RU δέχεται στην είσοδο αρχικά τα δεδομένα της νικητήριας αντίδρασης και στη συνέχεια τα δεδομένα των επηρεαζόμενων αντιδράσεων.
Initialize	Όταν είναι ενεργοποιημένο, η μονάδα RU δέχεται στην είσοδο τα δεδομένα όλων των αντιδράσεων του βιομοντέλου ώστε να πραγματοποιηθεί η αρχικοποίηση των εσωτερικών δομών του NPE.
Seed, SetSeed, MT_reset	Σήματα ελέγχου της μονάδας παραγωγής τυχαίων αριθμών, Mersenne Twister. Ενεργοποιούν την εγγραφή (Seed, SetSeed) ή την επαναφορά της μονάδας σε αρχική κατάσταση (MT_reset).
FG, K	Παράμετροι που επηρεάζουν τον υπολογισμό της τάσης $a_j$ της αντίδρασης.
$X_1, X_2, X_3$	Οι πληθυσμοί των αντιδρώντων της εκάστοτε αντίδρασης.
NReactions	Το πλήθος των αντιδράσεων του μοντέλου ( $m$ ).
NDReactions	Το πλήθος των επηρεαζόμενων αντιδράσεων της νικητήριας αντίδρασης ( $D$ ).
T_old, R_old_address	Ο χρόνος $\tau_{j,old}$ και η διεύθυνση ( $R_\mu, R_{\alpha 1}, \dots$ ) της εισερχόμενης αντίδρασης.
Tsim	Η τρέχουσα τιμή του χρόνου προσομοίωσης.
<b>Έξοδος</b>	
T_new, R_new_address	Ο ανανεωμένος χρόνος $\tau_{j,new}$ και η διεύθυνση ( $R_\mu, R_{\alpha 1}, \dots$ ) της εξερχόμενης αντίδρασης.
T_new_flag	Όταν είναι ενεργοποιημένο η μονάδα RU ενημερώνει το υψηλότερο επίπεδο της μονάδας NPE ότι στην έξοδο της μονάδας $T_{new}$ και $R_{new\_address}$ πρέπει να προωθηθούν στον πίνακα T.
Min_Tj	Ο μικρότερος χρόνος $\tau_{j,new}$ που προέκυψε από τους υπολογισμούς της RU.
Min_Tj_addr	Η διεύθυνση με τον μικρότερο χρόνο εκτέλεσης που δίνεται στην έξοδο.
Min_Tj_Rdy	Όταν το σήμα $Min\_Tj\_Rdy$ ενεργοποιηθεί τότε στην έξοδο ( $Min\_Tj, Min\_Tj\_addr$ ) έχει δοθεί η αντίδραση με τον μικρότερο χρόνο εκτέλεσης που υπολογίστηκε από τη μονάδα RU.

Το σχηματικό της Εικόνας 15 παρουσιάζει τη ροή των δεδομένων στη μονάδα RU. Η ροή των δεδομένων χωρίζεται σε τρεις διακλάδωσεις. Η πρώτη διακλάδωση (Branch A) αφορά τον υπολογισμό της τάσης μιας αντίδρασης ( $a_j$ , Propensity Unit) και τη διαίρεση των αριθμών ( $\frac{1}{a_j}, \frac{\alpha_{j,old}}{a_j}$ ). Η δεύτερη διακλάδωση (Branch B) αφορά μόνο τις επηρεαζόμενες αντιδράσεις και υπολογίζει τη διαφορά ( $\tau_{j,old} - t$ ). Τέλος, η τρίτη διακλάδωση (Branch C) υπολογίζει τον λογάριθμο ενός τυχαίου αριθμού ( $\ln\left(\frac{1}{r_j}\right)$ ) και ενεργοποιείται τόσο για την νικητήρια αντίδραση όσο και για κάθε επηρεαζόμενη αντίδραση που η προηγούμενη τιμή της τάσης  $a_{\alpha,old} = 0$ . Συγκεκριμένα, αν  $a_{\alpha,old} \neq 0$  τότε η RU επανυπολογίζει τον χρόνο ενεργοποίησης μιας επηρεαζόμενης αντίδρασης με βάση τον μαθηματικό τύπο (4) ενεργοποιώντας κατάλληλα την πρώτη και τη δεύτερη διακλάδωση, διαφορετικά με βάση τον μαθηματικό τύπο (3) ενεργοποιώντας κατάλληλα την πρώτη και τη

τρίτη διακλάδωση. Σε κάθε περίπτωση όλες οι έξοδοι των τριών διακλαδώσεων καταλήγουν μέσω πολυπλεξίας σε μια κοινή διακλάδωση (Common Branch) ώστε να ολοκληρωθούν όλες οι απαιτούμενες μαθηματικές πράξεις.

Στην έξοδο της μονάδας RU έχει τοποθετηθεί μια μονάδα σύγκρισης (Keep Min) έτσι ώστε όταν όλες οι πράξεις ολοκληρωθούν η μονάδα RU να έχει κρατήσει την αντίδραση με τον μικρότερο ανανεωμένο χρόνο. Το αποτέλεσμα της μονάδας RU θα συγκριθεί στη συνέχεια με το αποτέλεσμα του δέντρου συγκριτών, ενώ οι ανανεωμένοι χρόνοι  $\tau_{j,new}$  θα προωθηθούν προς αποθήκευση στον πίνακα T.



Εικόνα 15: Η αρχιτεκτονική της μονάδας Recalculation Unit.

Με βάση την αρχιτεκτονική της μονάδας RU (Εικόνας 15), οι περισσότερες μονάδες κινητής υποδιαστολής τοποθετούνται σειριακά, η μία μετά την άλλη, ώστε η έξοδος της μίας να είναι η είσοδος της επόμενης μονάδας. Γνωρίζοντας ότι κάθε μονάδα που εκτελεί αριθμητικές πράξεις με αριθμούς κινητής υποδιαστολής απαιτεί ένα συγκεκριμένο πλήθος κύκλων ρολογιού, καταλήγουμε στο εύλογο συμπέρασμα ότι η μονάδα RU απαιτεί ένα πολύ συγκεκριμένο πλήθος κύκλων ρολογιού για την ολοκλήρωση της διαδικασίας ανανέωσης του χρόνου μιας αντίδρασης.

Επίσης, το εργαλείο της Xilinx (Core Generator), που παράγει τις αριθμητικές μονάδες κινητής υποδιαστολής, μας δίνει τη δυνατότητα να αυξομειώσουμε τα επίπεδα σωλήνωσης (pipeline) και αντίστοιχα το απαιτούμενο πλήθος κύκλων ρολογιού (καθυστέρηση) της εκάστοτε μονάδας. Πρέπει να γίνει κατανοητό ότι ο βαθμός σωλήνωσης των μονάδων επιλέχθηκε με στόχο τον καλύτερο δυνατό συνδυασμό: τελικής συχνότητας λειτουργίας και απαιτούμενο πλήθος κύκλων ρολογιού. Οι δύο αυτές τιμές είναι αντιστρόφως ανάλογες, διότι όσο αυξάνονται τα επίπεδα σωλήνωσης στο εσωτερικό μιας μονάδας τόσο αυξάνεται η συχνότητα της μονάδας (ισχύει και το αντίστροφο). Η υπολογιστή ισχύς ενός συστήματος, όμως, δεν εξαρτάται μόνο από τη συχνότητα λειτουργίας του, αλλά, και από το πλήθος των κύκλων ρολογιού που απαιτούνται ώστε να ολοκληρωθούν οι πράξεις με τη μεγαλύτερη συχνότητα εκτέλεσης. Στη δικής μας περίπτωση, όλες πράξεις κινητής υποδιαστολής

εκτελούνται μέσα στη μονάδα RU.

Στον Πίνακα 8 που ακολουθεί παρουσιάζεται το πλήθος των απαιτούμενων κύκλων ρολογιού για κάθε μία από της αριθμητικές μονάδες που συνθέτουν τη μονάδα RU. Ο Πίνακας 8 είναι χωρισμένος με τέτοιο τρόπο ώστε να απεικονίζεται η συνολική καθυστέρηση για κάθε μια από τις διακλαδώσεις της RU, δίνοντας έμφαση στον συγχρονισμό των διακλαδώσεων.

Πίνακας 8: Οι απαιτούμενοι κύκλοι ρολογιού των διακλαδώσεων της RU.

	RU Latency								
BRANCH	Input	Branch Components Latency				Common Branch		Output	Total
A	Enable Register	Propensity		Div		Mult	Adder	Register	42
	1	14		17		4	5	1	
B	Enable Register	Sub		SR		Mult	Adder	Register	
	1	11		20		4	5	1	
C	Enable Register	Control Old $a_j$	MT	Fix2Float	Log	Mult	Adder	Register	
	1	4	1	6	20	4	5	1	

Συνεπώς, προκειμένου η RU να επαναυπολογίσει τον χρόνο ενεργοποίησης μιας αντίδρασης χρειάζεται 42 κύκλους ρολογιού. Αν  $D$  είναι το πλήθος των επηρεαζόμενων αντιδράσεων τότε ορίζουμε ως  $L_{RU} = 42 + D (cc)$  τη καθυστέρηση ώστε να ολοκληρωθούν όλες οι απαιτούμενες πράξεις από τη μονάδα RU. Τέλος, όπως αναφέρθηκε στην Ενότητα 6.3, είναι σημαντικό να τονίσουμε ότι με βάση τη τιμή  $L_{RU}$  και το πλήθος των αντιδράσεων ( $m$ ) ορίζουμε κατάλληλα το ύψος του δέντρου συγκριτών (CoT) ώστε να ισχύει  $L_{CoT} < L_{RU}$ , στοχεύοντας έτσι στην παράλληλη λειτουργία των μονάδων και στην ελαχιστοποίηση της συνολικής καθυστέρηση του NPE.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



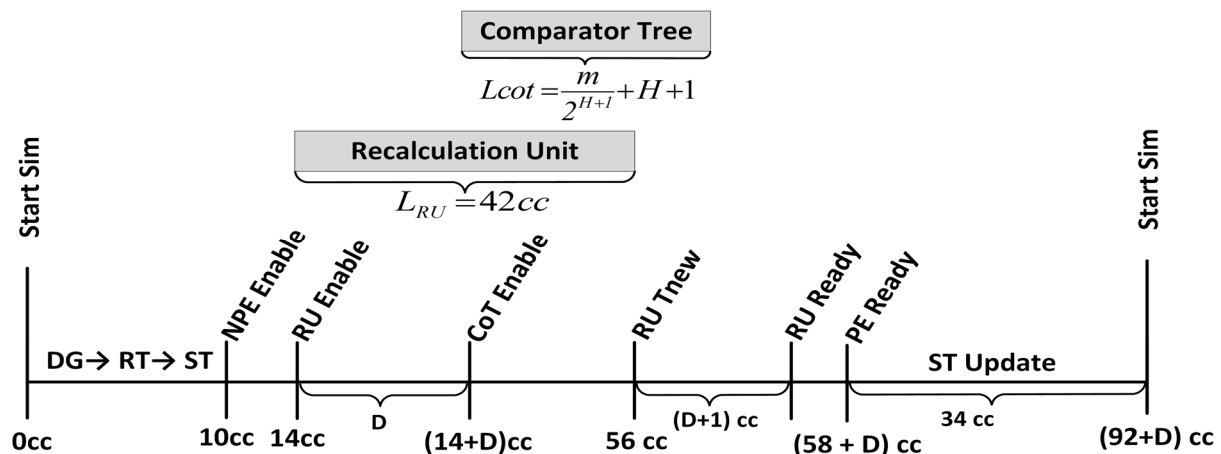
ΕΣΠΑ 2007-2013  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

## 7 ΕΠΙΔΟΣΕΙΣ - ΣΤΑΤΙΣΤΙΚΑ

### 7.1 Θεωρητική Ανάλυση Επιδόσεων

Στις προηγούμενες ενότητες έγινε εκτενής παρουσίαση των τεχνικών περιγραφών όλων των κύριων μονάδων του NRM πυρήνα και του NRM SoC. Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα της θεωρητικής ανάλυσης της καθυστέρησης λόγω διοχέτευσης (latency) της αρχιτεκτονικής, καθώς και πώς αυτή επηρεάζει την συνολική απόδοση του συστήματος.

Όπως έχει οριστεί σε προηγούμενη ενότητα, κατά την προσομοίωση ενός μοντέλου το σύστημα μεταβαίνει από την τρέχουσα χρονική στιγμή στην επόμενη με βάση το χρόνο ενεργοποίησης της νικητριάς αντιδράσεων (*winner reaction*), της αντίδρασης δηλαδή με το μικρότερο χρόνο ενεργοποίησης. Κάθε μετάβαση καθορίζεται στο τέλος ενός κύκλου αντίδρασης (*reactions cycle, RC*) μετά από συγκεκριμένο πλήθος κύκλων ρολογιού που είναι ανάλογο του πλήθους των επηρεαζόμενων αντιδράσεων ( $D$ ). Έχοντας μελετήσει διεξοδικά την αρχιτεκτονική της επεξεργαστικής μονάδας του NRM SoC είμαστε πλέον σε θέση να υπολογίσουμε την απαιτούμενη καθυστέρηση λόγω διοχέτευσης (latency) για να ολοκληρωθεί ένας κύκλος αντιδράσεων και να εκτιμήσουμε την απόδοση του συστήματος.



Εικόνα 16: Η ανάλυση της καθυστέρησης λόγω διοχέτευσης του πυρήνα NRM.

Όπως υποδεικνύει η Εικόνα 17, ένας κύκλος αντιδράσεων ξεκινάει με την ενεργοποίηση του σήματος *start\_sim* και ολοκληρώνεται με την εκκίνηση του επόμενου κύκλου αντίδρασης. Η ροή των δεδομένων ξεκινάει από τον γράφο εξαρτήσεων (DG), από όπου εξάγονται οι διευθύνσεις των επηρεαζόμενων αντιδράσεων, στη συνέχεια καταλήγουν στον πίνακα των αντιδράσεων (RT) και στον πίνακα των μοριακών ειδών (ST) από όπου εξάγονται όλα τα δεδομένα που συμμετέχουν στον επαναυπολογισμό του χρόνου ενεργοποίησης της εκάστοτε επηρεαζόμενης αντίδρασης.

Στη συνέχεια ενεργοποιείται η κεντρική επεξεργαστική μονάδα του πυρήνα (NPE), η οποία εκτελεί όλες τις αριθμητικές και λογικές πράξεις που χρειάζονται ώστε να προκύψει η επόμενη νικητριά αντίδραση. Όπως φαίνεται στο διάγραμμα της Εικόνας 17, αρχικά ενεργοποιείται η μονάδα RU προκειμένου να διαβαστούν και να στιγματιστούν οι  $D$  χρόνοι των αντιδράσεων από τον πίνακα **T**. Αφού αποδεσμευτεί ο πίνακας **T**, ενεργοποιείται το δέντρο συγκριτών CoT ώστε να βρεθεί η αντίδραση με το μικρότερο χρόνο  $\tau_j$  του πίνακα **T**. Το ύψος του δέντρου καθορίζεται αποκλειστικά από το πλήθος των αντιδράσεων ( $m$ ) ώστε να ισχύει  $L_{CoT} < L_{RU}$  (Ενότητα 6.3). Κατά αυτό τον τρόπο, το CoT ολοκληρώνει τις συγκρίσεις πριν η RU ολοκληρώσει τις αριθμητικές πράξεις, αποφεύγοντας έτσι την επιπλέον καθυστέρηση που απαιτείται για τον καθορισμό της

αντίδρασης με τον μικρότερο χρόνο ενεργοποίησης του πίνακα **T**. Συνεπώς, ορίζουμε ως καθυστέρηση (latency) της μονάδας NPE  $L_{NPE} = \max\{L_{RU}, L_{CoT}\} + D + 1 + 6 = L_{RU} + D + 7 (cc)$ , άρα  $L_{NPE} = 49 + D (cc)$ .

Όταν η μονάδα NPE υπολογίσει την επόμενη νικητήρια αντίδραση, τότε ο πυρήνας NRM μεταβαίνει στην κατάσταση ανανέωσης των πληθυσμών των μοριακών ειδών με βάση την στοιχειομετρία που ορίζει η νικητήρια αντίδραση. Για τον λόγο αυτό απαιτούνται επιπλέον 34 κύκλοι ρολογιού μέχρις ότου το σύστημα να είναι έτοιμο να ξεκινήσει τον επόμενο κύκλο αντίδρασης.

Εν κατακλείδι, ένας κύκλος αντίδρασης απαιτεί συνολικά  $L_{RC} = 92 + D$  κύκλους ρολογιού (cc) προκειμένου να καθοριστεί η επόμενη νικητήρια αντίδραση και το σύστημα να μεταβεί στην επόμενη κατάσταση. Στο σημείο αυτό πρέπει να τονίσουμε το γεγονός ότι ο τελικός απαιτούμενος χρόνος ενός κύκλου αντιδράσεων είναι ανεξάρτητος από το μέγεθος του μοντέλου ( $m$ ) και καθορίζεται από το πλήθος των επηρεαζόμενων αντιδράσεων ( $D$ ). Στην περίπτωση ενός πραγματικού μοντέλου, όπως το ASYN, το  $D$  είναι πιθανό να διαφέρει σε κάθε κύκλο αντίδρασης και για το λόγο αυτό ορίζουμε τον μέσο όρο επηρεαζόμενων αντιδράσεων  $D_{aver}$ .

Έχοντας αναλύσει την καθυστέρηση της κάθε μονάδας αλλά και τον συνολικό χρόνο καθυστέρησης ενός κύκλου αντιδράσεων, είμαστε πλέον σε θέση να εκτιμήσουμε την συνολική απόδοση του NRM Core. Δεδομένου ότι η μετάβαση του συστήματος από τη μια κατάσταση στην επόμενη εξαρτάται από την χρονική καθυστέρηση  $L_{RC}$ , χρησιμοποιούμε αυτό το μέγεθος για τη θεωρητική μελέτη της διεκπεραιωτικής ικανότητας (Reaction Throughput) του NRM Core:

- **Reaction Throughput (Mega Reactions Cycles / sec):** Το μέγεθος αυτό καθορίζει το πόσα εκατομμύρια κύκλοι αντιδράσεων (RCs) του δοθέντος μοντέλου ολοκληρώνονται σε ένα δευτερόλεπτο από το NRM Core (σε MRC/sec).

Για την εκτίμηση του MRC/sec θα πρέπει πρώτα να έχουν καθοριστεί τόσο το μέγεθος  $L_{RC}$  όσο και η συχνότητα λειτουργίας του συστήματος. Ο μαθηματικός τύπος (6) δίνει το Reaction Throughput ενός πυρήνα του συστήματος σε MRC/sec, αποδεικνύονται ότι εξαρτάται άμεσα από το πλήθος των επηρεαζόμενων αντιδράσεων ( $D_{aver}$ ) των RCs ενός βιομοντέλου και όχι από το συνολικό πλήθος των αντιδράσεων ( $m$ ):

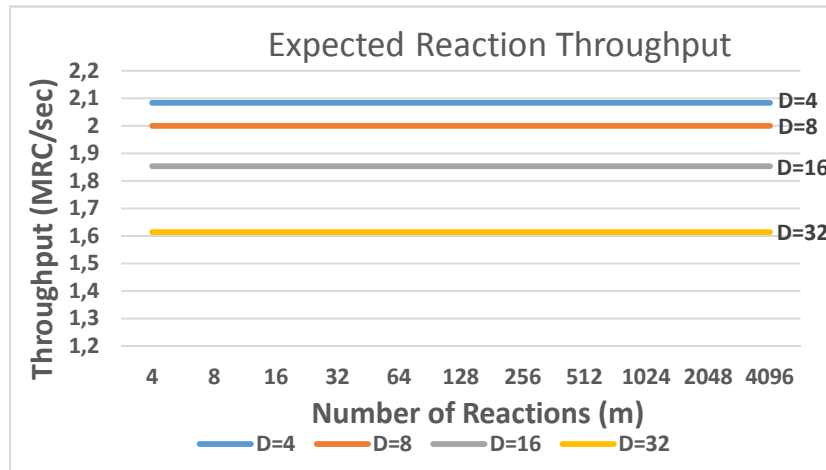
$$Reactions Throughput = \frac{1}{(92 + D_{aver}) \cdot t_{cycle} \cdot 10^6} (MRC/sec) \quad (6)$$

Στη συνέχεια ακολουθεί το διαγράμματα για την θεωρητική επίτευξη της διεκπεραιωτικής ικανότητας (Reaction Throughput) ενός NRM Core για διάφορους συνδυασμούς αριθμών αντιδράσεων του βιομοντέλου ( $m$ ) και μέσο όρο επηρεαζόμενων αντιδράσεων ( $D_{aver}$ ).

Όπως παρατηρούμε από το διάγραμμα της Εικόνας 17, με συχνότητα λειτουργίας τα 200MHz η διεκπεραιωτική ικανότητα (Reaction Throughput) του NRM Core δεν επηρεάζεται από το πλήθος των αντιδράσεων ( $m$ ) αλλά από το μέσο όρο των επηρεαζόμενων αντιδράσεων  $D_{aver}$  του εκάστοτε μοντέλου. Κατά αυτό τον τρόπο επιβεβαιώνεται η χρησιμότητα του NRM έναντι του FRM για βιομοντέλα με  $D_{aver} \ll m$ . Ωστόσο, είναι κατανοητό ότι όσο αυξάνεται το  $D_{aver}$  ομοίως αυξάνεται το μέγεθος  $L_{RC}$  και συνεπώς μειώνεται η διεκπεραιωτική ικανότητα του συστήματος. Τέλος, αν το NRM SoC περιλαμβάνει περισσότερους από έναν πυρήνες NRM τότε πολλαπλασιάζουμε τη διεκπεραιωτική ικανότητα με τον αριθμό των πυρήνων ( $C$ ) ώστε να προκύψει η συνολική διεκπεραιωτική ικανότητα του NRM SoC.

$$Total Reactions Throughput = C * \frac{1}{(92 + D_{aver}) \cdot t_{cycle} \cdot 10^6} (MRC/sec) \quad (7)$$





Εικόνα 17: Θεωρητικά αναμενόμενη διεκπεραιωτική ικανότητα (Reaction Throughput) του NRM SoC συναρτήσει του πλήθους των αντιδράσεων (m) και του μέσου όρου επηρεαζόμενων αντιδράσεων (Daver). Συχνότητα λειτουργίας 200MHz.

## 7.2 Στατιστικά Υλοποίησης

Στην ενότητα αυτή παρουσιάζονται τα στατιστικά υλοποίησης του NRM SoC με έναν ή περισσότερους πυρήνες για διάφορες τιμές των VHDL Generic παραμέτρων. Για τη σύνθεση και την υλοποίηση του συστήματος χρησιμοποιήθηκε το λογισμικό ISE 14.7 της εταιρίας Xilinx [29] και όλα τα στατιστικά που παρουσιάζονται σε αυτήν την ενότητα έχουν προκύψει από το στάδιο της υλοποίησης (Implementation). Η υλοποίηση του συστήματος έγινε με ένα μεσαίου μεγέθους FPGA, το XC7K325T-2FFG900 [30] της οικογένειας Kintex7 της εταιρίας Xilinx. Ακολουθεί ένας πίνακας με τα κύρια χαρακτηριστικά του FPGA XC7K325T.

Πίνακας 9: Χαρακτηριστικά του FPGA XC7K325T

XC7K325T				
Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	IOBs
407,600	203,800	445	840	500

Ακολουθούν οι πίνακες με τα στατιστικά υλοποίησης του NRM SoC με έναν πυρήνα ( $C = 1$ ) για διαφορετικές διαμορφώσεις με βάση τις γενικές παραμέτρους: πλήθος αντιδράσεων (m), πλήθος μοριακών ειδών (n) και μέσο όρο επηρεαζόμενων αντιδράσεων ( $D_{aver}$ ).

Πίνακας 10: Τα στατιστικά υλοποίησης του NRM SoC ( $C = 1$ ) για  $m = n = 512$  και διαφορετικά  $D_{aver}$ .

NRM SoC ( $C = 1$ ) $m = n = 512$ ( $H = 3$ )					
$D_{aver}$	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	16,062 (3%)	15,726 (7%)	36 (7.5%)	33 (3%)	240
8	16,151 (3%)	15,793 (7%)	37 (8%)	33 (3%)	230
16	16,644 (4%)	16,266 (7%)	39 (8.5%)	33 (3%)	230
32	17,278 (4%)	16,892 (8%)	43 (9%)	33 (3%)	220

Πίνακας 11: Τα στατιστικά υλοποίησης του NRM SoC ( $C = 1$ ) για  $m = n = 1024$  και διαφορετικά  $D_{aver}$ .

NRM SoC ( $C = 1$ ) $m = n = 1024$ ( $H = 4$ )					
$D_{aver}$	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	16,388 (4%)	16,208 (8%)	45 (9.5%)	33 (3%)	230
8	16,997 (4%)	16,445 (8%)	46 (10%)	33 (3%)	220
16	17,319 (4%)	16,816 (8%)	48 (10.5%)	33 (3%)	220
32	19,156 (4%)	18,338 (9%)	53 (11%)	33 (3%)	220

Πίνακας 12: Τα στατιστικά υλοποίησης του NRM SoC ( $C = 1$ ) για  $m = n = 2048$  και διαφορετικά  $D_{aver}$ .

NRM SoC ( $C = 1$ ) $m = n = 2048$ ( $H = 5$ )					
$D_{aver}$	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	17,701 (4%)	17,684 (9%)	74 (17%)	33 (3%)	210
8	17,805 (4%)	17,756 (9%)	77 (17%)	33 (3%)	200
16	18,317 (4%)	18,123 (9%)	82 (18.5%)	33 (3%)	200
32	19,633 (4%)	19,411 (10%)	92 (21%)	33 (3%)	200

Πίνακας 13: Τα στατιστικά υλοποίησης του NRM SoC ( $C = 1$ ) για  $m = n = 4096$  και διαφορετικά  $D_{aver}$ .

NRM SoC ( $C = 1$ ) $m = n = 4096$ ( $H = 6$ )					
$D_{aver}$	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	20,471 (5%)	20521 (10%)	135 (30%)	33 (3%)	200
8	20,550 (5%)	20709 (10%)	140 (31%)	33 (3%)	190
16	21,192 (5%)	21236 (10%)	151 (34%)	33 (3%)	190
32	22,475 (5%)	22357 (11%)	173 (39%)	33 (3%)	170



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



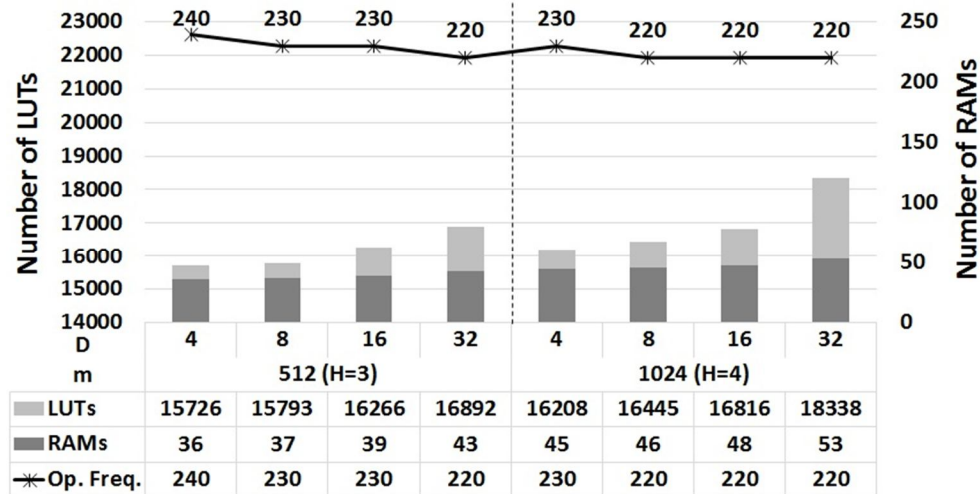
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
επένδυση στην κοινωνία της γνώσης  
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

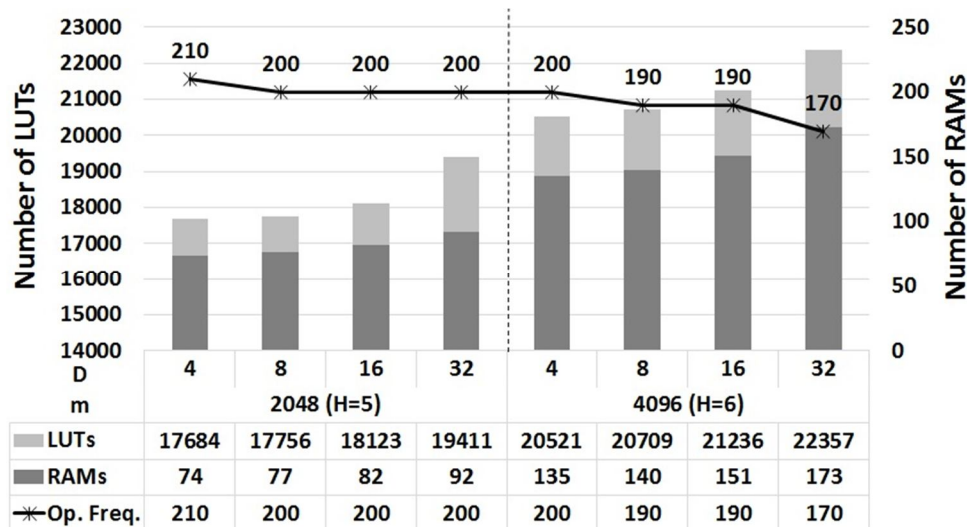


ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Ακολουθούν διαγράμματα με τα στατιστικά υλοποίησης του NRM SoC με έναν πυρήνα ( $C = 1$ ) όπως αυτά παρουσιάζονται στους προηγούμενους πίνακες. Από τα διαγράμματα παραλείπουμε το πλήθος των DSPs διότι παραμένει σταθερό (33 – 3%) καθώς και το πλήθος των Flip-Flops δεδομένου ότι ακολουθούν παρόμοια κλιμάκωση με τα LUTs.



Εικόνα 18: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM Core για  $m = 512/1024$  και διαφορετικά  $D_{aver}$ .



Εικόνα 19: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM Core για  $m = 2048/4096$  και διαφορετικά  $D_{aver}$ .

Μελετώντας τα παραπάνω διαγράμματα των διαφορετικών υλοποιήσεων διαπιστώνουμε την κλιμάκωση των στατιστικών καθώς αυξάνονται οι παράμετροι  $m$ ,  $n$  και  $D_{aver}$ . Χαρακτηριστικό είναι επίσης το γεγονός ότι διατηρώντας σταθερό το πλήθος των αντιδράσεων ( $m$ ) και αυξάνοντας το πλήθος των επηρεαζόμενων αντιδράσεων ( $D_{aver}$ ) παρουσιάζεται μείωση της συχνότητας λειτουργίας του NRM – SoC. Η μείωση αυτή οφείλεται στην αύξηση του μεγέθους του γράφου εξαρτήσεων και στον τρόπο με τον οποίο το εργαλείο

σύνθεσης συνδυάζει τις BRAMs ώστε να παράγει το επιθυμητό μέγεθος του γράφου. Όπως έχει αναφερθεί στην Ενότητα 5.3, ο γράφος εξαρτήσεων αποθηκεύεται με τη μορφή πίνακα του οποίου το πλάτος είναι  $(D_{aver} + 3) * \log(m)$  και επηρεάζεται άμεσα από τη παράμετρο  $D_{aver}$ . Επομένως, όσο αυξάνεται η παράμετρος  $D_{aver}$ , περισσότερες BRAMs συνδυάζονται, προκαλώντας προβλήματα δρομολόγησης (routing problems) και επηρεάζοντας έτσι τη συχνότητα λειτουργίας του συστήματος.

Ακολουθούν οι πίνακες με τα στατιστικά υλοποίησης του NRM SoC με το μέγιστο δυνατό αριθμό από πυρήνες για διαφορετικές διαμορφώσεις του συστήματος με βάση τις γενικές παραμέτρους: πλήθος αντιδράσεων ( $m$ ), πλήθος μοριακών ειδών ( $n$ ) και μέσο όρο επηρεαζόμενων αντιδράσεων ( $D_{aver}$ ). Το πλήθος των πυρήνων εξαρτάται από τις απαιτήσεις του συστήματος σε πόρους και κατά πόσο είναι δυνατόν να καλυφθούν στο διαθέσιμο FPGA.

Πίνακας 14: Τα στατιστικά υλοποίησης του NRM SoC ( $C = 8$ ) για  $m = n = 512$  και διαφορετικά  $D_{aver}$ .

NRM SoC ( $C = 8$ ) $m = n = 512$ ( $H = 3$ )					
$D_{aver}$	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	92,857 (22%)	99,221 (48%)	201 (45%)	33 (3%)	200
8	98,081 (24%)	102,906 (50%)	209 (47%)	33 (3%)	200
16	99,303 (24%)	104,213 (51%)	230 (51.5%)	33 (3%)	200
32	101,745 (24%)	106,625 (52%)	257 (58%)	33 (3%)	190

Πίνακας 15: Τα στατιστικά υλοποίησης του NRM SoC ( $C = 8$ ) για  $m = n = 1024$  και διαφορετικά  $D_{aver}$ .

NRM SoC ( $C = 8$ ) $m = n = 1024$ ( $H = 4$ )					
$D_{aver}$	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	109,569 (26%)	108,593 (53%)	269 (60.5%)	33 (3%)	180
8	110,297 (27%)	109,341 (53%)	281 (63%)	33 (3%)	160
16	111,687 (27%)	111,250 (54%)	297 (67%)	33 (3%)	170
32	114,257 (28%)	113,619 (55%)	333 (74.5%)	33 (3%)	170

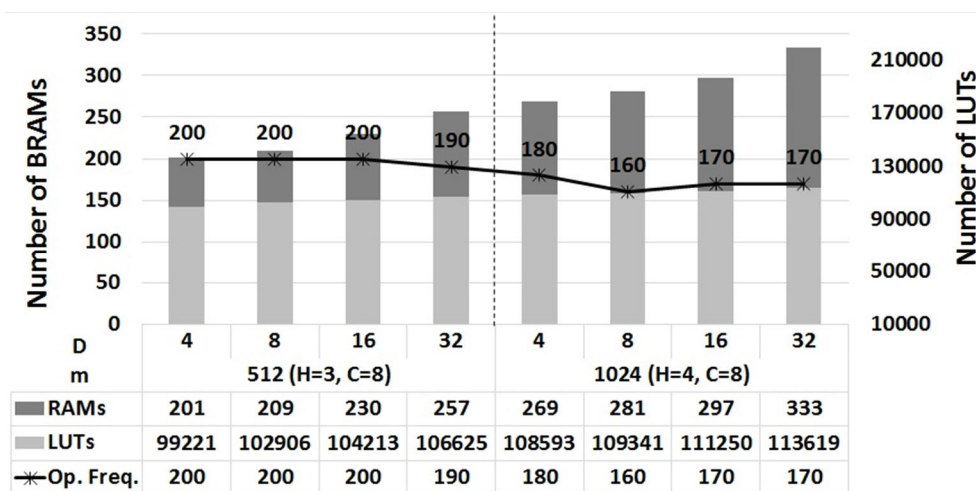
Πίνακας 16: Τα στατιστικά υλοποίησης του NRM SoC ( $C = 4$ ) για  $m = n = 2048$  και διαφορετικά  $D_{aver}$ .

NRM SoC ( $C = 4$ ) $m = n = 2048$ ( $H = 5$ )					
$D_{aver}$	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	58,838 (14%)	65,999 (32%)	261 (58.5%)	33 (3%)	190
8	64,948 (15%)	59,386 (29%)	271 (61%)	33 (3%)	200
16	59,890 (14%)	66,835 (32%)	291 (65%)	33 (3%)	180
32	61,238 (15%)	68,095 (33%)	329 (74%)	33 (3%)	180

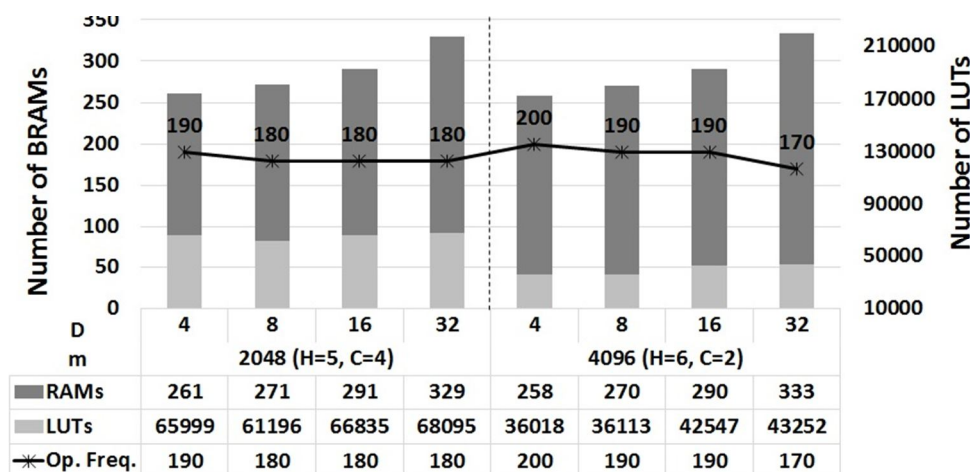
Πίνακας 17: Τα στατιστικά υλοποίησης του NRM SoC ( $C = 2$ ) για  $m = n = 4096$  και διαφορετικά  $D_{aver}$ .

NRM SoC ( $C = 2$ ) $m = n = 4096$ ( $H = 6$ )					
$D_{aver}$	Flip-Flops	LUTs	BRAMs (36Kbit)	DSPs	Freq. (MHz)
4	36,573 (8%)	36,018 (17%)	258 (58%)	33 (3%)	200
8	36,727 (9%)	36,113 (17%)	270 (61%)	33 (3%)	190
16	40,439 (9%)	42,547 (20%)	290 (65%)	33 (3%)	190
32	41,519 (10%)	43,252 (21%)	333 (75%)	33 (3%)	170

Ακολουθούν διαγράμματα με τα στατιστικά υλοποίησης του NRM SoC για διαφορετικές τιμές των παραμέτρων  $m$ ,  $n$ ,  $D_{aver}$  και στο μέγιστο αριθμό πυρήνων ( $C$ ) ανά υλοποίηση.



Εικόνα 20: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM SoC για  $m = 512/1024$  και διαφορετικά  $D_{aver}$ .



Εικόνα 21: Το διάγραμμα με τα στατιστικά υλοποίησης του NRM SoC για  $m = 2048/4096$  και διαφορετικά  $D_{aver}$ .



Στα διαγράμματα των Εικόνων 20 και 21 απεικονίζονται οι απαιτήσεις του NRM SoC για διαφορετικές υλοποιήσεις με βάση τις επιλεγμένες γενικές παραμέτρους. Εύλογα συμπεραίνουμε ότι όσο μικρότερες τιμές επιλέγουμε για τις παραμέτρους  $m$  και  $n$  τόσο περισσότερους πυρήνες μπορούμε να ενσωματώσουμε στο σύστημα. Ενώ, επίσης, είναι λογικό να παρουσιάζεται αυξομείωση της συχνότητας λειτουργίας καθώς αυξάνονται οι πυρήνες ή αυξάνεται η παράμετρος  $D_{aver}$ . Όπως αναφέρθηκε και προηγουμένως, η πυκνότητα του τελικού σχεδίου προκαλεί προβλήματα δρομολόγησης είτε λόγω των απαιτήσεων σε επιφάνεια του FPGA (Slices, LUTs) είτε λόγω των απαιτήσεων σε μνήμη (BRAMs).



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
Επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
Πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

## 8 ΕΠΙΚΥΡΩΣΗ ΣΧΕΔΙΑΣΗΣ

Στις προηγούμενες ενότητες παρουσιάστηκε η αρχιτεκτονική NRM SoC και τα στατιστικά υλοποίησης για διαφορετικούς συνδυασμούς των παραμέτρων. Στην παρούσα ενότητα, θα παρουσιαστεί η μεθοδολογία που ακολουθήθηκε για τον έλεγχο και την επικύρωση της λειτουργίας της αρχιτεκτονικής. Επίσης θα παρουσιαστούν τα αποτελέσματα των δοκιμών που διεξήχθησαν χρησιμοποιώντας βιομοντέλα τύπου θηράματος/θηρευτή.(Lotka Volterra System – LVS [41]).

Κατά τη σχεδίαση ενός πολύπλοκου υπολογιστικού συστήματος, όπως το NRM SoC, ο έλεγχος της λειτουργικότητας (design verification) και ο έλεγχος της ορθότητας (validation) του συστήματος παίζουν ιδιαίτερα σημαντικό ρόλο. Η διαδικασία ελέγχου της λειτουργικότητας πραγματοποιείται τόσο κατά τη σχεδίαση των επιμέρους μονάδων του συστήματος όσο και στο τελικό ολοκληρωμένο σύστημα. Στην περίπτωση μας χρησιμοποιήσαμε το λογισμικό προσομοίωσης iSim [42] της εταιρίας Xilinx, εκτελώντας αρχικά προσομοιώσεις λογικής συμπεριφοράς (behavioral simulation [43]) του κυκλώματος και στη συνέχεια προσομοιώσεις χρονισμού σε επίπεδο πύλης (timing simulation [44]). Κατά αυτό τον τρόπο, διευκολύνεται η σχεδίαση και ο συγχρονισμός των πολύπλοκων συστημάτων, και αποφεύγονται σημαντικές αστοχίες στο τελικό ενοποιημένο σχέδιο.

Αφού ολοκληρώθηκε ο έλεγχος της λειτουργικότητας του NRM SoC εκτελώντας μια πληθώρα από κατάλληλα προγράμματα επαλήθευσης (VHDL testbenches), καλούμαστε να εκτελέσουμε ένα πραγματικό πείραμα προσομοίωσης που να επικυρώνει την ορθότητα του συστήματος. Όπως και στο Παραδοτέο 1.1 για το FRM SoC έτσι και στην περίπτωση του NRM SoC εκτελέσαμε την προσομοίωση για ένα γνωστό βιομοντέλο που γνωρίζουμε την συμπεριφορά του.

Η επαλήθευση των αποτελεσμάτων πραγματοποιείται συγκρίνοντας τα αποτελέσματα του NRM SoC με τα αποτελέσματα ενός δημοφιλή προσομοιωτή, το λογισμικό COPASI [45], εκτελώντας προσομοίωση για το βιομοντέλο LVS (Lotka Volterra System) του οποίου τα χαρακτηριστικά είναι τα εξής:

- Αρχικοί πληθυσμοί μοριακών ειδών:
  - Θηρευτής (Predator): 100
  - Θήραμα (Prey): 100
- Σταθερές ρυθμού αντίδρασης ( $K$ ) για τις τρεις αντιδράσεις του βιομοντέλου LVS:
  - $K_1$ : 0,5
  - $K_2$ : 0,0025
  - $K_3$ : 0,33

Ακολουθούν οι παράμετροι της προσομοίωσης του βιομοντέλου LVS:

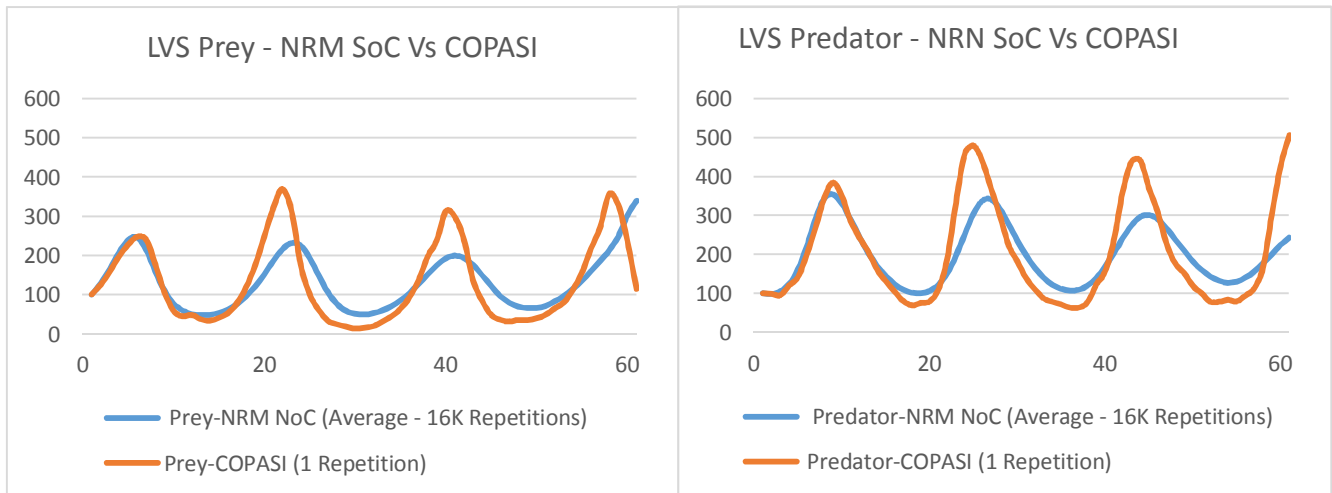
- $T_{sim}$ : 60 s (διάρκεια προσομοίωσης)
- $T_{sam}$ : 1 s (περίοδος δειγματοληψίας αποτελεσμάτων)
- $R$ : 16K (στοχαστικές επαναλήψεις)
- Max Internal Steps: 1000K

Και, τέλος, οι γενικές παράμετροι του NRM SoC όπως αυτές ορίστηκαν πριν τη σύνθεση του συστήματος και οι οποίες καλύπτουν στο έπακρον τις ανάγκες της προσομοίωσης:

- $C = 16$  NRM Cores
- $m = 128$
- $n = 128$

- $H = 1$
- $D_{aver} = 4$
- NRM NoC freq.: 210 MHz

Για την μεγιστοποίηση της απόδοσης του συστήματος, χρησιμοποιούμε 16 NRM πυρήνες, όπου κάθε πυρήνας είναι ανεξάρτητος, χρησιμοποιεί τη δική του αλληλουχία τυχαίων αριθμών και στη συγκεκριμένη περίπτωση εκτελεί 1000 επαναλήψεις της προσομοίωσης. Τέλος, εκτελούμε το ίδιο πείραμα στο COPASI χρησιμοποιώντας τις ίδιες αρχικές συνθήκες και παραμέτρους. Ακολουθούν τα διαγράμματα των αποτελεσμάτων του NRM SoC και του COPASI.



Εικόνα 22: Σύγκριση των προσομοιωτών NRM NoC και COPASI

Στην Εικόνα 22 συγκρίνουμε τα αποτελέσματα του NRM SoC με τα αντίστοιχα αποτελέσματα του COPASI, για τη συμπεριφορά του «θηράματος» (prey) στο αριστερό μέρος και τη συμπεριφορά του «θηρευτή» (predator) στο δεξί μέρος. Και στις δύο περιπτώσεις η συμπεριφορά των μοριακών ειδών είναι σχεδόν πανομοιότυπη, αποδεικνύοντας την ορθότητα της στοχαστικής προσομοίωσης που εκτελεί το NRM SoC.

Τέλος, συγκρίνουμε τον πραγματικό χρόνο εκτέλεσης της προσομοίωσης του NRM SoC και του COPASI, θέλοντας έτσι να δείξουμε την αύξηση της απόδοσης του συστήματός μας έναντι της σειριακής εκτέλεσης του πειράματος από τον προσομοιωτή COPASI. Στο NRM SoC η εκτέλεση  $R = 16K$  στοχαστικών επαναλήψεων διήρκεσε  $T_{real} = 6.01$  δευτερόλεπτα (sec) πραγματικού χρόνου, ενώ ο προσομοιωτής COPASI χρειάστηκε  $T_{real} = 116$  δευτερόλεπτα (sec) για το ίδιο πείραμα. Ο προσομοιωτής COPASI εκτέλεσε τις 16K προσομοιώσεις σε επεξεργαστή τελευταίας γενιάς (Intel i7) με συχνότητα λειτουργίας 2,4 GHz ενώ είχε στη διάθεση του 32GB μνήμης RAM. Στη δοκιμή με το βιομοντέλο LVS το σύστημα NRM SoC με 16 πυρήνες στα 210 MHz αποδεικνύεται 19.3 φορές ταχύτερο (speedup factor) σε σύγκριση με τον προσομοιωτή COPASI.

Στη συνέχεια του έργου έχουμε προγραμματίσει επιπλέον δοκιμές με δημοσιευμένα μοντέλα από την Biomodels Database. Επίσης θα διεξαχθούν δοκιμές και με μοντέλα benchmark, με ελεγχόμενη και κλιμακούμενη πολυπλοκότητα, με σκοπό την μελέτη των επιδόσεων που επιτυγχάνονται στη πράξη από την αρχιτεκτονική NRM SoC και την σύγκριση των αποτελεσμάτων αυτών τόσο με τη θεωρητική εκτίμηση των επιδόσεων της Ενότητας 7.1 όσο και με τα αντίστοιχα αποτελέσματα του FRM SoC για τα αντίστοιχα πειράματα.

## 9 ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Στα πλαίσια της παρούσης έρευνας, σχεδιάσαμε και αναπτύξαμε επιτυχώς ένα Σύστημα σε Ψηφίδα (SoC) που να προσομοιώνει αποδοτικά πολύπλοκα δίκτυα βιομοριακών αλληλεπιδράσεων που περιγράφουν τη συμπεριφορά κυτταρικών υποσυστημάτων, υλοποιώντας τον δημοφιλή αλγόριθμο στοχαστικής προσομοίωσης Next Reaction Method (NRM) των Gibson και Bruck. Σε όλη τη διάρκεια της σχεδίασης, της υλοποίησης και της επικύρωσης του συστήματος ήρθαμε αντιμέτωποι με πληθώρα από προκλήσεις, οι οποίες όμως δεν αποτέλεσαν εμπόδιο στην επιτυχή ολοκλήρωση του συστήματος. Είμαστε σε θέση να δηλώσουμε ότι το σύστημα αυτό αποτελεί διεθνή καινοτομία και ξερνά σε επιδόσεις και δυνατότητες κλιμάκωσης τις υπάρχουσες λύσεις. Σχεδιάζουμε την υποβολή προς δημοσίευση των αποτελεσμάτων που παρουσιάστηκαν.

Κατά το στάδιο της σχεδίασης, δεδομένου ότι ο αλγόριθμος NRM είναι εγγενώς σειριακός προβληματιστήκαμε ιδιαίτερα στο να βρούμε τον πλέον κατάλληλο τρόπο να επωφεληθούμε από τις δυνατότητες που παρέχει μια υλοποίηση υλικού. Για τον λόγο αυτόν, επικεντρωθήκαμε ιδιαίτερα στη πλήρη διοχέτευση (pipelining) για τη μείωση της καθυστέρησης (latency) του διαύλου επεξεργασίας (datapath) που απαιτείται για την εύρεση της επόμενης νικητριάς αντίδρασης, χρησιμοποιώντας μονάδες παράλληλης σύγκρισης δεδομένων (CoT), και υλοποιώντας τις μαθηματικές πράξεις με υψηλής απόδοσης μονάδες (RU). Δόθηκε ιδιαίτερη σημασία στη παράλληλη λειτουργία αλλά και στο συγχρονισμό των μονάδων CoT και RU με τρόπο που να επιτρέπει, μέσω της επιλογής του ύψους  $H$  του δένδρου, η σταθερή καθυστέρηση της RU να "κρύβει" (latency hiding) τη καθυστέρηση του CoT, ώστε να ελαχιστοποιούνται οι απαιτούμενοι κύκλοι ρολογιού ενός κύκλου αντιδράσεων (RC latency) και να αποφεύγεται η χρήση επιπλέον μονάδων ελέγχου. Εν τέλει, το σύστημά μας είναι σε θέση να εκτελεί μεγάλης κλίμακας βιομοντέλα χωρίς να επηρεάζεται η απόδοσή του από το συνολικό πλήθος των αντιδράσεων ( $m$ ) αλλά να εξαρτάται αποκλειστικά και μόνο από το μέσο όρο των επηρεαζόμενων αντιδράσεων ( $D_{aver}$ ) του βιομοντέλου, κάτι που αποτελεί καινοτόμο προσέγγιση στο πρόβλημα.

Σημαντικό ζήτημα είναι και η εύρεση κατάλληλων μοντέλων που η πολυπλοκότητά τους να βασίζεται στην κλιμάκωση του μέσου όρου των επηρεαζόμενων αντιδράσεων ( $D_{aver}$ ) για σταθερό συνολικό αριθμό αντιδράσεων ( $m$ ). Τέτοια βιομοντέλα θα βοηθήσουν στο να επιδείξουμε και πρακτικά τη κλιμάκωση της διεκπεραιωτικής ικανότητας του συστήματος, όπως αυτή έχει αναλυθεί θεωρητικά στην Ενότητα 7.1. Στη βιβλιογραφία δεν υπάρχουν βιομοντέλα-benchmarks που να πληρούν αυτές τις προϋποθέσεις, άρα διερευνούμε τη συστηματική δημιουργία τους από την ομάδα μας με αναδρομικό τρόπο, κάτι που θα αποτελέσει και γενικότερη συνεισφορά στο χώρο.

Επιπλέον, έχοντας υπόψη την πολυπλοκότητα των δύο αλγορίθμων (FRM, NRM) αλλά και την τελική απόδοση των συστημάτων που αναπτύξαμε για κάθε έναν από αυτούς, μας ενδιαφέρει να μελετήσουμε τη βέλτιστη επιλογή αλγορίθμου με βάση την πολυπλοκότητα του βιομοντέλου: πλήθος αντιδράσεων ( $m$ ) και μέσος όρος επηρεαζόμενων αντιδράσεων ( $D_{aver}$ ). Σκοπός μας είναι να μπορούμε να εκτελούμε στοχαστική προσομοίωση επιλέγοντας τον πλέον κατάλληλο αλγόριθμο για δοθέν βιομοντέλο με στόχο την μεγιστοποίηση της απόδοσης. Αυτό θα γίνεται φυσικά αυτόματα από το σύστημα μας και χωρίς ο τελικός χρήστης να χρειάζεται να αναλύσει τα χαρακτηριστικά της εργασίας προσομοίωσης που επιθυμεί να εκτελέσει προκειμένου να αποφασίσει την βέλτιστη λύση.

Τέλος, μελετώντας τα στατιστικά υλοποίησης (Ενότητα 7.2) καταλήγουμε στο συμπέρασμα ότι με ένα μεγαλύτερο FPGA το σύστημά μας θα είναι σε θέση να καλύψει μοντέλα με πλήθος αντιδράσεων μεγαλύτερο

ακόμα και από 4096 ( $m > 4096$ ). Με βάση τη σημερινή τεχνολογία το XC7K325T θεωρείται ένα μετρίου μεγέθους FPGA, γεγονός που περιορίζει την περαιτέρω κλιμάκωση του NRM SoC. Συνεπώς, κατά το στάδιο της σύνθεσης του NRM SoC μας απασχόλησε ιδιαίτερα το μέγιστο επιτρεπόμενο πλήθος αντιδράσεων και κατά πόσο επηρεάζεται η συχνότητα λειτουργίας του συστήματος όσο αυξάνονται οι απαιτούμενοι πόροι. Πρόκειται για ένα ενδιαφέρον ζήτημα που χρήζει περαιτέρω διερεύνησης, χρησιμοποιώντας μεγαλύτερου μεγέθους και καλύτερης τεχνολογίας FPGAs. Παρόλο που η συχνότητα μειώνεται ελφάρως όσο το σχέδιο μεγαλώνει, η μείωσή της δεν είναι δραστική λόγω της επιθετικής διοχέτευσης (pipelining) που χρησιμοποιήθηκε στη ανάπτυξη του πλήρως παραμετρικού σχεδίου.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
*επένδυση στην κοινωνία της γνώσης*  
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Wikipedia Contributors, «Biology,» Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., 4 2 2015. [Ηλεκτρονικό]. Available: [www.en.wikipedia.org/wiki/Biology](http://www.en.wikipedia.org/wiki/Biology).
- [2] Wikipedia Cotributors, «Template: Branches of Biology,» Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., [Ηλεκτρονικό]. Available: [www.e.wikipedia.org/wiki/Template:Branches\\_of\\_biology](http://www.e.wikipedia.org/wiki/Template:Branches_of_biology). [Πρόσβαση 4 2 2015].
- [3] F. Stanley, «The interplay of biology and technology,» *Proceedings of the National Academy of Science*, τόμ. 98, αρ. 18, pp. 10051-10054, 2001.
- [4] Wikipedia contributors, «Computational biology,» Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., 07-Jun-2012. [Ηλεκτρονικό].
- [5] Wikipedia contributors, «Bioinformatics,» Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., 07-Jun-2012. [Ηλεκτρονικό].
- [6] Wikipedia contributors, «Data mining,» Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., 08-Jun-2012. [Ηλεκτρονικό].
- [7] Wikipedia contributors, «Control theory,» Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., 29-May-2012. [Ηλεκτρονικό].
- [8] Wikipedia contributors, «Network analysis (electrical circuits),» Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., 29-May-2012. [Ηλεκτρονικό].
- [9] Wikipedia contributors, «Artificial intelligence,» Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., 7 Jun 2012. [Ηλεκτρονικό]. [Πρόσβαση Feb 2015].
- [10] Wikipedia contributors, «Pattern recognition,» Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., 29-May-2012. [Ηλεκτρονικό].
- [11] A. Finkelstein, J. Hetherigton, O. Margoninski, P. Saffrey, R. Seymour και A. Warner, «Computational challenges of systems biology,» *Computer*, τόμ. 37, 2004.
- [12] A. P. Heath και L. E. Kavradi, «Computational challenges in systems biology,» *Computer Science Review*, τόμ. 3, αρ. 1, pp. 1-17, 2009.
- [13] M. Tomita, «Whole-cell simulation: a grand challenge of the 21st century,» *TRENDS in Biotechnology*, τόμ. 19, pp. 205-210, 1 June 2001.
- [14] D. T. Gillespie, «Stochastic simulation of chemical kinetics,» *Annu. Rev. Phys. Chem.*, τόμ. 58, pp. 35-55, 2007.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
Επένδυση στην κοινωνία της γνώσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

- [15] D. J. Wilkinson, Stochastic modelling for systems biology, CRC press, 2011.
- [16] Wikipedia Contributors, «Markov Chain,» Wikipedia, the free encyclopedia. Wikimedia Foundation, Inc., [Ηλεκτρονικό]. Available: [www.en.wikipedia.org/wiki/Markov\\_chain](http://www.en.wikipedia.org/wiki/Markov_chain). [Πρόσβαση 4 2 2015].
- [17] M. A. Gibson και J. Bruck, «Efficient exact stochastic simulation of chemical systems with many species and many channels,» *The Journal of Physical Chemistry A*, τόμ. 104, pp. 1876-1889, Feb 2000.
- [18] Y. Masato, I. Yow, N. Yuri, K. Toshinori, O. Yasunori, F. Akira, S. Yuichiro, I. Naoki, Y. Hideki, K. Hiroaki και A. Hideharu, «FPGA Impementation of a Data-Driven Stoachastic Biochemical Simulator with the Next Reaction Method,» *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, pp. 254-259, 2007.
- [19] M. Yoshimi, Y. Nishikawa, Y. Osana, A. Funahaski, Y. Shibata, H. Yamada, N. Hiroi, K. Hiroaki και H. Amano, «Practical implementation of a network-based stochastic biochemical simulation system on an FPGA,» *Field Programmable Logic and Applications*, pp. 663-666, Sept 2008.
- [20] D. B. Thomas και H. Amano, «A fully pipelined FPGA architecture for stochastic simulation of chemical systems,» *Field Programmable Logic and Applications*, pp. 1-7, Sept 2013.
- [21] O. Hazapi και E. Manolakos, «Scalable FRM-SSA SoC design for the simulation of networks with thousands of biochemical reactions in real time,» *Field Programmable Logic and Applications (FPL), 2011 International Conference on*, pp. 459-463, 2011.
- [22] H. Yamada, Y. Osana, T. Ishimori, T. Ooya, M. Yoshimi, Y. Nishikawa, A. Funahashi, N. Horoi, H. Amano και Y. Shibata, «A Modular Approach to Heterogeneous Biochemical Model Simulation on an FPGA,» *Reconfigurable Computing and FPGAs, 2009. ReConFig'09. International Conference on*, pp. 125-130, Dec 2009.
- [23] M. Yoshimi, Y. Osana, Y. Iwaoka, Y. Nishikawa, T. Kojima, N. Hiroi, Y. Shibata, N. Iwanaga και H. Kitano, «An FPGA Implementation of High Throughput Stochastic Simulator for Large-Scale Biochemical Systems,» *Field Programmable Logic and Applications, 2006. FPL'06. International Conference on*, pp. 1-6, Aug 2006.
- [24] N. Azizi, I. Kuon, A. Egier, A. Darabiha και P. Chow, «Reconfigurable molecular dynamics simulator,» *Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on*, pp. 197-206, April 2004.
- [25] J. F. Keane, C. Bradley και C. Ebeling, «A compiled accelerator for biological cell signaling simulations,» *Proceedings of the 2004 ACM/SIGDA 12th International Symposium on Field Programmable Gate Arrays*, pp. 233-241, 2004.
- [26] D. T. Gillespie, «Exact stochastic simulation of coupled chemical reactions,» *The Journal of Physical Chemistry*, τόμ. 81, pp. 2340-2361, 1977.
- [27] C. J. Myers, Engineering genetic circuits, CRC Press, 2011.

- [28] E. Ouzounoglou, D. Kalamatianos, E. Emmanouilidou, M. Xilouri, L. Stefanis, K. Vekrellis και E. S. Manolakos, «In silico modeling of the effects of alpha-synuclein oligomerization on dopaminergic neuronal homeostasis,» *BMC Systems Biology*, τόμ. 8, αρ. 1, p. 54, May 2014.
- [29] Xilinx, «Xilinx,» Xilinx Inc, [Ηλεκτρονικό]. Available: <http://www.xilinx.com/>. [Πρόσβαση 01 Jun 2013].
- [30] Xilinx, «Xilinx Inc,» [Ηλεκτρονικό]. Available: [http://www.xilinx.com/publications/prod\\_mktg/Kintex7-Product-Table.pdf](http://www.xilinx.com/publications/prod_mktg/Kintex7-Product-Table.pdf). [Πρόσβαση 21 5 2015].
- [31] Xilinx, «ISE Design Suite,» Xilinx Inc, [Ηλεκτρονικό]. Available: <http://www.xilinx.com/products/design-tools/ise-design-suite.html>. [Πρόσβαση 01 Jul 2011].
- [32] Xilinx, «Running the Generate Programming File Process for FPGAs,» Xilinx Inc, [Ηλεκτρονικό]. Available: [http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx11/ise\\_p\\_generate\\_fpga\\_programming\\_file.htm](http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/ise_p_generate_fpga_programming_file.htm). [Πρόσβαση 01 Jun 2013].
- [33] Xilinx, «iMpat Overview,» Xilinx Inc, [Ηλεκτρονικό]. Available: [http://www.xilinx.com/itp/xilinx10/isehelp/pim\\_c\\_overview.htm](http://www.xilinx.com/itp/xilinx10/isehelp/pim_c_overview.htm). [Πρόσβαση 01 Jul 2012].
- [34] Vermeulen, Bart and Waayers, Tom and Bakker και Sjaak, «IEEE 1149.1-compliant access architecture for multiple core debug on digital system chips,» *Test Conference, 2002. Proceedings. International*, pp. 55-63, 2002.
- [35] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. Doyle, H. Kitano, A. Arkin, B. Bornstein, D. Bray και A. Cornish-Bowden, «The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models,» *Bioinformatics*, τόμ. 19, pp. 524-531, 2003.
- [36] Wikipedia contributors, «Comma-separated values,» Wikipedia, the free encyclopedia.Wikimedia Foundation, Inc. [Ηλεκτρονικό]. [Πρόσβαση 01 Jul 2011].
- [37] A. AMBA, «Specification Rev 2.0,» *ARM Limited*, 1999.
- [38] T. Henriksson και P. Van der Wolf, «Ttl hardware interface: A high-level interface for streaming multiprocessor architectures,» *Proceedings of the 2006 IEEE/ACM/IFIP Workshop on Embedded Systems for Real Time Multimedia*, pp. 107-112, Oct 2006.
- [39] M. Matsumoto και T. Nishimura, «Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator,» *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, τόμ. 8, αρ. ACM, pp. 3-30, Jan. 1998.
- [40] N. H. Beebe, «A Bibliography of IEEE Standards,» 2013.
- [41] M. V. Schneider, *In Silico Systems Biology*, Humana Press, 2013.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ

ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
Επένδυση στην κοινωνία της γνώσης



ΕΣΠΑ  
2007-2013  
Πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

- [42] Xilinx, «ISE Simulator (ISim),» Xilinx Inc, [Ηλεκτρονικό]. Available: <http://www.xilinx.com/tools/isim.htm>. [Πρόσβαση 01 Jun 2013].
- [43] Xilinx, «Behavioral Simulation,» Xilinx Inc, [Ηλεκτρονικό]. Available: [http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx11/platform\\_studio/ps\\_c\\_sim\\_behavioral\\_simulation.htm](http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/platform_studio/ps_c_sim_behavioral_simulation.htm). [Πρόσβαση 01 Jun 2013].
- [44] Xilinx, «Timing Simulation,» Xilinx Inc, [Ηλεκτρονικό]. Available: [http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx11/ise\\_c\\_simulation\\_timing.htm](http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/ise_c_simulation_timing.htm). [Πρόσβαση 01 Jun 2013].
- [45] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, S. Natalia, M. Singhal, L. Xu, P. Mendes και U. Kummer, «COPASI—a complex pathway simulator,» *Bioinformatics*, τόμ. 22, pp. 3067-3074, 2006.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ  
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ  
Επένδυση στην κοινωνία της γνώσης

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
Πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ